

Attribute Efficient Linear Regression with Data-Dependent Sampling

Doron Kukliasnky
Weizmann Institute of Science
doronk@weizmann.ac.il

Ohad Shamir
Weizmann Institute of Science
ohad.shamir@weizmann.ac.il

Abstract

In this paper we analyze a budgeted learning setting, in which the learner can only choose and observe a small subset of the attributes of each training example. We develop efficient algorithms for ridge and lasso linear regression, which utilize the geometry of the data by a novel data-dependent sampling scheme. When the learner has prior knowledge on the second moments of the attributes, the optimal sampling probabilities can be calculated precisely, and result in data-dependent improvements factors for the excess risk over the state-of-the-art that may be as large as $O(\sqrt{d})$, where d is the problem's dimension. Moreover, under reasonable assumptions our algorithms can use *less* attributes than full-information algorithms, which is the main concern in budgeted learning settings. To the best of our knowledge, these are the first algorithms able to do so in our setting. Where no such prior knowledge is available, we develop a simple estimation technique that given a sufficient amount of training examples, achieves similar improvements. We complement our theoretical analysis with experiments on several data sets which support our claims.

1 Introduction

Linear regression is a longstanding and effective method for learning a prediction model from various data sets. However, in some scenarios, it cannot be utilized as-is: The algorithms that solve this problem assume they can access all the attributes of the training set, whereas there are some real-life scenarios where the learner can access only a small number of attributes per training example.

Consider, for example, the problem of medical diagnosis in which the learner wishes to determine whether a patient has some disease based on a series of medical tests. In order to build a linear model, the learner has to gather a set of volunteers, perform diagnostic tests on them and use the tests results as features. However, some of the volunteers may be reluctant to undergo a large number of tests, as medical tests may cause physical discomfort, and will prefer to undergo only a small number of them. During test time, however, patients are more likely to agree to undergo all tests, to find a diagnosis to their illness.

Another example is the case where there is some cost associated with each attribute, whether computational or financial. For example, the <http://intelligence.towerdata.com> web site allows users to buy marketing data about email addresses and pay per feature. The learner would like to minimize the cost, which is not necessarily the number of examples.

This problem is known as budgeted learning [1] or learning with limited attribute observation (LAO) [2]. Formally, we use the local budget setting presented in [3]: For each training example

(composed of a d -dimensional attribute vector \mathbf{x} and a target value y), we have a budget of $k + 1$ attributes, where $k \ll d$, and we are able to choose which attribute we wish to reveal. This is different from the missing data setting, in which the selected attributes are given to us, and we are not able to choose which attributes to reveal, and from the feature selection setting, in which the output model includes only a subset of the attributes. In our setting, the goal is to find a good predictor despite the partial information at training time where a good predictor is defined as one that minimizes the expected discrepancy between the predicted value, \hat{y} , and the target value, y . This discrepancy is generally measured by some kind of loss function, and we focus on the squared loss i.e. $\ell(\hat{y}, y) = \frac{1}{2}(\hat{y} - y)^2$. The expected discrepancy over the training set is called the risk.

We consider learning with respect to linear predictors, parameterized by a vector $\mathbf{w} \in \mathbb{R}^d$. Given an unlabeled example \mathbf{x} (a vector of attributes), the prediction is defined as $\langle \mathbf{w}, \mathbf{x} \rangle^1$. In particular, we focus on two standard types of linear prediction problems: Those with L_2 bounded norm, which are the ridge regression scenario; and those with L_1 bounded norm, which are the lasso regression scenario.

Our basic approach is similar to the one proposed in [4, 3], which uses online gradient descent with stochastic gradient estimates. The general idea behind it is to scan through the training set, calculate an unbiased gradient estimator based on each example (using only a small number of attributes), and plug it into a stochastic gradient descent method, thus minimizing the loss over the training set.

The algorithms in [4, 3] build the unbiased estimator using uniform sampling from the attributes of the example, eventually leading to ridge algorithms with expected excess risk bounds of $O(\sqrt{d/km})$ after m examples, compared with $O(\sqrt{1/m})$ for the online full-information algorithms that can view all the attributes [5], and lasso algorithms with an additional $\log d$ factor for both settings (see Table 1). Another interpretation of these results is that when viewing only k out of d attributes, the algorithms need $O(d/k)$ times as many examples to obtain the same accuracy, thus examining the same number of attributes. [4] also provides a lower bound for the ridge scenario establishing that the ridge bound is not improvable in general.

In this paper, despite these seemingly unimprovable results, we show that they can in fact be improved. We do this by developing a novel sampling scheme which samples the attributes in a data-dependent manner: We sample attributes with large second moments more than others, thus are able to gain a *data-dependent* improvement factor. In other words, our sampling methods take advantage of the geometry of the data distribution, and utilize it to extract more 'information' out of each sample. Under reasonable assumptions, our methods need to examine *less* attributes to reach the same accuracy than the online full-information algorithms, thus optimizing the principal goal in budgeted scenarios. To the best of our knowledge, ours are the first methods able to do so in the local budget setting.

We begin by assuming prior knowledge of the second moments of the data, namely $\mathbb{E}_D[\mathbf{x}_i^2]$ for $i \in [d]$, where we use $\mathbb{E}_D[\cdot]$ to denote the expectation with respect the data distribution. Our risk bounds, under the assumptions of $\|\mathbf{x}\|_2 \leq 1$ in the ridge scenario and $\|\mathbf{x}\|_\infty \leq 1$ in the lasso scenario are also summarized in Table 1. To clarify the notation, $\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}}$ is defined as

$$\left(\sum_{i=1}^d \sqrt{\mathbb{E}_D[\mathbf{x}_i^2]}\right)^2, \text{ and } \|\mathbb{E}_D[\mathbf{x}^2]\|_1 \text{ is defined as } \sum_{i=1}^d \mathbb{E}_D[\mathbf{x}_i^2].$$

It can be easily shown that both $\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}}$ and $\|\mathbb{E}_D[\mathbf{x}^2]\|_1$ are smaller than or equal to d ,

¹We ignore the bias term here, but this can be easily handled by adding a constant dummy attribute that will always be revealed.

	New Bound	Old Bound	Online Full-Information Bound
Ridge Regression	$O\left(\sqrt{\left(\ \mathbb{E}_D[\mathbf{x}^2]\ _{\frac{1}{2}} + k\right)/km}\right)$	$O\left(\sqrt{d/km}\right)$	$O\left(\sqrt{1/m}\right)$
Lasso Regression	$O\left(\sqrt{\left(\ \mathbb{E}_D[\mathbf{x}^2]\ _1 + k\right)\log d/km}\right)$	$O\left(\sqrt{d\log d/km}\right)$	$O\left(\sqrt{\log d/m}\right)$

Table 1: The expected excess risk bounds of the various algorithms under the assumptions of $\|\mathbf{x}\|_2 \leq 1$ in the ridge scenario and $\|\mathbf{x}\|_\infty \leq 1$ in the lasso scenario.

which proves that our bounds are always as good as the previous bounds. In fact, the equalities hold only when all the moments are exactly the same. Otherwise, both values are strictly smaller than d , making our bounds better than the previous. This improvement factor is data-dependent and may be as large as $O(\sqrt{d})$ as both values can be small as $O(1)$, when the moments decay at a sufficient rate. In fact, similar distributional assumptions are made in other successful algorithmic approaches such as AdaGrad (we further elaborate on the connection between our work and AdaGrad in Appendix A). When the attribute budget satisfies $k = \Omega\left(\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}}\right)$ (or $k = \Omega\left(\|\mathbb{E}_D[\mathbf{x}^2]\|_1\right)$ in the lasso scenario) our bounds also coincide with the online full-information scenario.

Of course, a practical limitation of our approach is that the second moments of the data may not be known in advance or easily computable in our attribute efficient setting. To address this, we split our algorithms into two phases: In the first phase, we use a simple yet effective estimation scheme that estimates the second moments of the attributes. In the second phase, we use the same sampling scheme but with smoothed probabilities, to compensate for the stochastic nature of the estimation phase. We prove that this method is always as good as the previous algorithms (up to constant factors) and given sufficient training examples, achieves the same bounds as our algorithms with prior knowledge on the second moments of the attributes (up to constant factors).

The rest of this paper is organized as follows: In section 2 we provide necessary background. In section 3 we describe the existing state of the art algorithms for attribute efficient ridge regression, and develop our sampling scheme for the case where we have prior knowledge of the second moments of the attributes. We also develop an estimation scheme for the case where we do not assume any prior knowledge of the second moments of the attributes, and present two variants of the algorithm: one that does assume prior knowledge of $\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}}$ only, and one that does not assume any prior knowledge at all. These two variants have the same expected risk bounds (up to a constant factor), but differ by the number of training examples needed in the estimation phase. In section 4 we provide similar results, this time for attribute efficient lasso regression. When no prior knowledge of the second moment of the attributes is available, the lasso scenario is simpler than the ridge scenario, as prior knowledge of $\|\mathbb{E}_D[\mathbf{x}^2]\|_1$ does not improve the results. In section 5 we show experimental results that support our theoretical claims, both on simulated and on well known data sets. We finish with a summary in section 6, and short discussion about the connection between the AdaGrad method and our sampling scheme in appendix A.

2 Preliminaries

2.1 Notation

Throughout this paper we use the following notations: We indicate scalars by a small letter, a , and vectors by a bold font, \mathbf{a} . We use \mathbf{a}^2 to indicate the vector for which $\mathbf{a}^2[i] = a[i]^2$ for all i , and $\mathbf{a} + b$ to indicate the vector for which $(\mathbf{a} + b)[i] = a[i] + b$. We denote the i -th vector of the standard basis by \mathbf{e}_i . All our vectors lie in \mathbb{R}^d , where d is the dimension. We indicate the set of indices $1, \dots, n$ by $[n]$. We use $\|\mathbf{a}\|_p$ to indicate the p -norm of the vector, equal to $\left(\sum_{i=1}^d |a_i|^p\right)^{\frac{1}{p}}$. We apply this notation also for the case where $p = \frac{1}{2}$ i.e. $\|\mathbf{a}\|_{\frac{1}{2}} = (\sum_{i=1}^d \sqrt{|a_i|})^2$, even though this is not a proper norm, as the triangle inequality does not hold. We also use $\|\mathbf{a}\|_{\infty}$ to indicate the infinity norm, $\max_i |a_i|$. We use $\langle \mathbf{a}, \mathbf{b} \rangle$ to indicate the standard inner product, $\sum_{i=1}^d a_i b_i$. We denote the expectation with respect to the randomness of the algorithm (attribute sampling) by $\mathbb{E}_A[\cdot]$, the expectation with respect to the data distribution by $\mathbb{E}_D[\cdot]$ and the expectation with respect to both by $\mathbb{E}_{D,A}[\cdot]$. For the two-phased algorithms, we use $\mathbb{E}_{D,A_i}[\cdot]$ where $i \in \{1, 2\}$ to denote the expectation with respect to the data distributions and the randomness of the algorithm during the i -th phase.

2.2 Linear Regression

The general framework for regression assumes the learner has a training set: $\{(\mathbf{x}_t, y_t) \in \mathbb{R}^d \times \mathbb{R}\}_{t=1}^m$, where each \mathbf{x}_t is a data point, represented by a vector of attributes, and y_t is the desired target value. The goal of the learner is to find a weight vector \mathbf{w} , such that $\hat{y}_t = \langle \mathbf{w}, \mathbf{x}_t \rangle$ is a good estimator of y_t , in the sense that it minimizes some penalty function over the entire data set. We focus on the most popular choice for such a function - the squared loss: $\ell(\hat{y}, y) = \frac{1}{2}(\hat{y} - y)^2$. We denote the loss induced by (\mathbf{x}_t, y_t) as $\ell_t(\mathbf{w})$.

We follow the standard framework for statistical learning [6] and assume the training set was sampled i.i.d. from some joint distribution \mathcal{D} . The goal of the learner is to find a predictor that minimizes the risk, defined as the expected loss:

$$L_{\mathcal{D}}(\mathbf{w}) = \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}} [\ell(\mathbf{w}^T \mathbf{x}, y)].$$

Since the distribution \mathcal{D} is unknown, the learner relies on the given training set, $S = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)\}$ that is assumed to be sampled i.i.d. from \mathcal{D} .

Finding this minimum may result in over fitting the data, therefore it is common to limit the size of the hypothesis class by adding some regularization constraint on the norm of \mathbf{w} , requiring it to be smaller than or equal to some value. The first of the two main scenarios of regression is ridge regression, where we have the 2-norm constraint, and the hypothesis class is $\mathcal{F} = \{\mathbf{w} | \|\mathbf{w}\|_2 \leq B\}$. If we assume $|y_t| \leq B$, using the Cauchy-Schwarz inequality, we can assume without loss of generality that $\|\mathbf{x}\|_2 \leq 1$ with probability 1. The second is lasso regression, where we have the 1-norm constraint, and the hypothesis class is $\mathcal{F} = \{\mathbf{w} | \|\mathbf{w}\|_1 \leq B\}$. Since we assume $|y_t| \leq B$, using the Hölder inequality, we can assume without loss of generality that $\|\mathbf{x}\|_{\infty} \leq 1$ with probability 1.

In the full-information regression scenario, the learner has access to all the attributes of \mathbf{x}_t , whereas in the attribute efficient scenario, the learner can sample at most $k + 1$ attributes out of d from each vector \mathbf{x}_t .

2.3 Related Work

The scenario of learning with limited attribute access was first introduced by Ben-David & Dichterman [2], under the term "Learning with Restricted Focus of Attention". There are two popular types of constraints: The first, which we address in this paper, is the local budget constraint, where the learner has access to $k + 1$ attributes per training example. The second is the global budget constraint where the learner has access to a total number of K attributes, and may spread them freely among all the training examples, as long as the total number of attributes seen does not exceed K . Clearly, any upper bound for the local budget setting holds also in the global budget setting for $K = (k + 1)m$.

Cesa-Bianchi et al. in [3] were the first to build an efficient linear algorithm for the local budget scenario, and asked the question of whether there exists an efficient algorithm for the attribute efficient scenario that can reach a similar accuracy as the full attribute scenario, while seeing $O(md/k)$ examples and from each example being able to sample only $O(k)$ attributes. Such a result would imply that in the attribute efficient scenario, we can learn just as well as in the full-information scenario, after seeing the same number of attributes ($O(md)$ in both cases). Thus, we can trade-off between the number of examples and the amount of information received on each example. They also proved a lower sample complexity bound of $\Omega(d/k\epsilon)$ examples for learning an ϵ -accurate linear regressor.

Later on, Hazan et al. [4] showed that the answer is yes, up to global constants for both the ridge and lasso scenarios. Their approach for ridge regression was based on the Online Gradient Descent method [7] and on the EG algorithm [8] for the lasso scenario. In both cases, at each iteration, the learner uses an unbiased estimator of the gradient, and updates the current weight vector accordingly. The key idea is that by sampling just a few attributes using an appropriate scheme, the learner can still build an unbiased estimator of the gradient, even in the attribute efficient scenario, and by expectation, perform a gradient step in the correct direction. They also complemented the ridge regression result by proving a corresponding lower sample complexity bound of $\Omega(d/k\epsilon^2)$ examples for learning an ϵ -accurate ridge regressor.

3 Attribute Efficient Ridge Regression

In this section we present our algorithms for ridge regression, where the loss is the squared loss: $\ell(\mathbf{w}; \mathbf{x}_t, y_t) = \frac{1}{2} (\langle \mathbf{w}, \mathbf{x}_t \rangle - y_t)^2$, and the 2-norm is bounded, $\|\mathbf{w}\|_2 \leq B$. The generic approach to the ridge attribute efficient scenario, which we call the General Attribute Efficient Ridge Regression (GAERR) algorithm and is presented in Algorithm 1, was first developed in [3, 4] and is based on the Online Gradient Descent (OGD) algorithm with gradient estimates.

The OGD algorithm goes over the training set, and for each example builds an unbiased estimator of the gradient. Afterwards, the algorithm updates the current weight vector, \mathbf{w}_t , by performing a step of size η in the opposite direction to the gradient estimator. The result is projected over the L_2 ball of size B , yielding \mathbf{w}_{t+1} . At the end, the algorithm outputs the average of all \mathbf{w}_t . The algorithm converges to the global minimum, as the minimization problem is convex in \mathbf{w} .

The gradient of the squared loss is $\nabla \ell(\mathbf{w}; \mathbf{x}_t, y_t) = (\langle \mathbf{w}, \mathbf{x}_t \rangle - y_t) \cdot \mathbf{x}_t$, and the key idea of the GAERR algorithm is how to use the budgeted sampling to construct an unbiased estimator for the gradient. The GAERR algorithm does so by sampling $k + 1$ attributes out of the d attributes of the

sample where $k > 0$ is the a budget parameter²: First, it samples k attributes with probabilities q_i and by weighting them correctly, builds an unbiased estimator for the data point $\tilde{\mathbf{x}}_t$. Then it samples one attribute with probability $p_{j_t} = \frac{w_{t,j_t}^2}{\|\mathbf{w}_t\|_2^2}$ and by a simple calculation obtains an unbiased estimator of the inner product. Reducing the label, y_t , yields the unbiased estimator, $\tilde{\phi}_t$. Finally, the algorithms multiplies the estimator of the inner product minus the label, $\tilde{\phi}_t$, by the estimator of the data point, $\tilde{\mathbf{x}}_t$, thus building an unbiased estimator of the gradient for the point, $\tilde{\mathbf{g}}_t$.

Algorithm 1 GAERR

Parameters: $B, \eta > 0$ and q_i for $i \in [d]$

Input: training set $S = \{(\mathbf{x}_t, y_t)\}_{t \in [m]}$ and $k > 0$

Output: regressor $\bar{\mathbf{w}}$ with $\|\bar{\mathbf{w}}\|_2 \leq B$

- 1: Initialize $\mathbf{w}_1 \neq 0$, $\|\mathbf{w}_1\|_2 \leq B$ arbitrarily
 - 2: **for** $t = 1$ to m **do**
 - 3: **for** $r = 1$ to k **do**
 - 4: Pick $i_{t,r} \in [d]$ with probability $q_{i_{t,r}}$ and observe $\mathbf{x}_t[i_{t,r}]$
 - 5: $\tilde{\mathbf{x}}_{t,r} \leftarrow \frac{1}{q_{i_{t,r}}} \mathbf{x}_t[i_{t,r}] \mathbf{e}_{i_{t,r}}$
 - 6: **end for**
 - 7: $\tilde{\mathbf{x}}_t \leftarrow \frac{1}{k} \sum_{r=1}^k \tilde{\mathbf{x}}_{t,r}$
 - 8: Choose $j_t \in [d]$ with probability $p_{j_t} = \frac{w_{t,j_t}^2}{\|\mathbf{w}_t\|_2^2}$ and observe $\mathbf{x}_t[j_t]$
 - 9: $\tilde{\phi}_t \leftarrow \frac{w_{t,j_t}}{p_{j_t}} \mathbf{x}_t[j_t] - y_t$
 - 10: $\tilde{\mathbf{g}}_t \leftarrow \tilde{\phi}_t \cdot \tilde{\mathbf{x}}_t$
 - 11: $\mathbf{v}_t \leftarrow \mathbf{w}_t - \eta \tilde{\mathbf{g}}_t$
 - 12: $\tilde{\mathbf{w}}_{t+1} \leftarrow \mathbf{v}_t \cdot \frac{B}{\max\{\|\mathbf{v}_t\|_2, B\}}$
 - 13: **end for**
 - 14: $\bar{\mathbf{w}} \leftarrow \frac{1}{m} \sum_{t=1}^m \mathbf{w}_t$
-

²As in the AERR algorithm, we assume we have a budget of at least 2 attributes per training sample.

The expected risk bound of the GAERR algorithm is presented in the next theorem which is a slightly more general version of Theorem 3.1 in [4].

Theorem 3.1. *Assume the distribution \mathcal{D} is such that $\|\mathbf{x}\|_2 \leq 1$ and $|y| < B$ with probability 1. Let $\bar{\mathbf{w}}$ be the output of GAERR when run with step size η and let $\max_t \mathbb{E}_{D,A} [\|\tilde{\mathbf{g}}_t\|_2^2] \leq G^2$. Then for any $\mathbf{w}^* \in \mathbb{R}^d$ with $\|\mathbf{w}^*\|_2 \leq B$,*

$$\mathbb{E}_{D,A} [L_{\mathcal{D}}(\bar{\mathbf{w}})] \leq L_{\mathcal{D}}(\mathbf{w}^*) + \frac{2B^2}{\eta m} + \frac{\eta}{2} G^2.$$

The general idea of the proof is that $\tilde{\mathbf{g}}_t$ is an unbiased estimator of the gradient, therefore we can use the standard analysis of the OGD algorithm. The full proof can be found in appendix B.1.

The AERR algorithm is one variant of the GAERR algorithm. It was presented in [4] and uses uniform sampling to estimate \mathbf{x}_t . In our GAERR notation it uses

$$q_i = \frac{1}{d} \quad \forall i \in [d].$$

The authors prove (Lemma 3.3 in [4]) that for the AERR algorithm, $G^2 \leq 8B^2 d/k$, which together with Theorem 3.1 and using $\eta = \frac{2B}{G\sqrt{m}}$ yields an expected risk bound of $4B^2 \sqrt{\frac{2d}{km}}$. They also prove that up to constant factors, their algorithm is optimal, by showing a corresponding lower bound.

This, however, is not the end of the story. By analyzing the bound, we show that we can improve the bound in a data-dependent manner. Theorem 3.1 shows us that the expected risk bound is proportional to G , therefore we wish to develop a sampling method that minimizes the 2-norm of the gradient estimator.

The gradient estimate consists of estimating the inner product and estimating \mathbf{x}_t . To estimate \mathbf{x}_t , we use the following procedure: we sample k indices, $i_{t,r}$, from $1..d$ with probability q_i , and use

$$\tilde{\mathbf{x}}_t = \frac{1}{k} \sum_{r=1}^k \frac{1}{q_{i_{t,r}}} \mathbf{x}_t[i_{t,r}] \mathbf{e}_{i_{t,r}} \quad (1)$$

as an estimator for \mathbf{x}_t . The next lemma will assist in bounding its 2-norm.

Lemma 3.2. *For every distribution (q_1, \dots, q_d) where $q_i \geq 0$ and $\sum_{i=1}^d q_i = 1$, we have $\mathbb{E}_{D,A} [\|\tilde{\mathbf{x}}_t\|_2^2] = \frac{1}{k} \mathbb{E}_{D,A} [\|\tilde{\mathbf{x}}_{t,r}\|_2^2] + \frac{k-1}{k} \mathbb{E}_D [\|\mathbf{x}\|_2^2]$.*

The proof can be found in appendix B.2.

Since

$$\mathbb{E}_{D,A} [\|\tilde{\mathbf{x}}_{t,r}\|_2^2] = \mathbb{E}_{D,A} [\tilde{\mathbf{x}}_{t,r}[i_{t,r}]^2] = \sum_{i=1}^d \frac{1}{q_i} \mathbb{E}_D [x_i^2], \quad (2)$$

in order to minimize the 2-norm of the estimator, we need to solve the following optimization problem:

$$\begin{aligned} & \underset{q_i}{\text{minimize}} && \frac{1}{k} \sum_{i=1}^d \frac{1}{q_i} \mathbb{E}_D [x_i^2] + \frac{k-1}{k} \mathbb{E}_D [\|\mathbf{x}\|_2^2] \\ & \text{subject to} && \sum_{i=1}^d q_i = 1, \quad \forall i \quad q_i \geq 0. \end{aligned}$$

This problem is equivalent to

$$\begin{aligned} & \underset{q_i}{\text{minimize}} && \sum_{i=1}^d \frac{1}{q_i} \mathbb{E}_D [x_i^2] \\ & \text{subject to} && \sum_{i=1}^d q_i = 1, \forall i \ q_i \geq 0, \end{aligned} \tag{3}$$

which can easily be solved using the Lagrange multipliers method to yield the solution

$$q_i = \frac{\sqrt{\mathbb{E}_D [x_i^2]}}{\sum_{j=1}^d \sqrt{\mathbb{E}_D [x_j^2]}}. \tag{4}$$

We also use

$$\tilde{\phi}_t = \frac{w_{t,j}}{p_{j_t}} \mathbf{x}_t[j_t] - y_t \tag{5}$$

as an estimator for the inner product minus the label. The next lemma will assist in bounding its 2-norm.

Lemma 3.3. *Using our sampling method we have $\mathbb{E}_{D,A} [\tilde{\phi}_t^2] \leq 4B^2$.*

The proof can be found in appendix B.3.

We could have followed a similar optimization strategy for finding the optimal sampling distribution for estimating the inner product. This strategy would have yielded that the optimal probabilities are $p_i = \frac{\sqrt{\mathbf{w}_{t,i}^2 \mathbb{E}_D [x_i^2]}}{\sum_{j=1}^d \sqrt{\mathbf{w}_{t,j}^2 \mathbb{E}_D [x_j^2]}}$. We, however, were not able to prove the superiority of this sampling method analytically and it was left out of the algorithm analysis.

Altogether, we formulate a lemma that will bound the gradient estimate.

Lemma 3.4. *The GAERR algorithm generates gradient estimates that for all t , $\mathbb{E}_{D,A} [\|\tilde{\mathbf{g}}_t\|_2^2] \leq 4B^2 \left(\frac{1}{k} \mathbb{E}_{D,A} [\|\tilde{\mathbf{x}}_{t,r}\|_2^2] + 1 \right)$.*

Proof. This lemma follows directly from Lemmas 3.2 and 3.3, using the independence of $\tilde{\mathbf{x}}_t$ and $\tilde{\phi}_t$ given \mathbf{x}_t and $\|\mathbf{x}\|_2 \leq 1$. \square

3.1 Known Second Moment Scenario

If we assume we have prior knowledge of the second moment of each attribute, namely $\mathbb{E}_D [x_i^2]$ for all $i \in [d]$, we can use equation (4) to calculate the optimal values of the q_i -s. This is the idea behind our DDAERR (Data-Dependent Attribute Efficient Ridge Regression) algorithm.

The expected risk bound of our algorithm is formulated in the next theorem.

Theorem 3.5. *Assume the distribution \mathcal{D} is such that $\|\mathbf{x}\|_2 \leq 1$ and $|y| \leq B$ with probability 1 and $\mathbb{E}_D [x_i^2]$ are known for $i \in [d]$. Let $\bar{\mathbf{w}}$ be the output of DDAERR, when run with $\eta = \frac{1}{\sqrt{m \left(\frac{1}{k} \|\mathbb{E}_D [\mathbf{x}^2]\|_{\frac{1}{2}} + 1 \right)}}$. Then for any $\mathbf{w}^* \in \mathbb{R}^d$ with $\|\mathbf{w}^*\|_2 \leq B$,*

$$\mathbb{E}_{D,A} [L_{\mathcal{D}}(\bar{\mathbf{w}})] \leq L_{\mathcal{D}}(\mathbf{w}^*) + 4 \frac{B^2}{\sqrt{m}} \sqrt{\frac{1}{k} \|\mathbb{E}_D [\mathbf{x}^2]\|_{\frac{1}{2}} + 1}.$$

Proof of Theorem 3.5. The theorem follows directly from Theorem 3.1, Lemma 3.4, equation (2) and the calculated q_i -s in equation (4). \square

Recalling that with probability 1 we have $\|\mathbf{x}\|_2 \leq 1$, it is easy to see that $\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}} \leq d$, therefore the DDAERR algorithm always performs at least as well as the AERR algorithm³. However, $\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}}$ may also be much smaller than d , in cases where the second moments varies between attributes or the vector is sparse. In these cases, we may gain a significant improvement. For example, if we consider a polynomial attribute decay such as $\mathbb{E}_D[x_i^2] = \frac{i^{-2}}{\sum_{j=1}^d j^{-2}}$, we have $\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}} = O(\log^2 d)$ which is significantly smaller than d .

3.2 Unknown Second Moment Scenario, Known $\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}}$

The solution presented in the previous section requires exact knowledge of $\mathbb{E}_D[x_i^2]$ for all i . Such prior knowledge may not be available when the learner is faced with a new learning task. Thus, we turn to consider the case where the moments are initially unknown. We will still assume that the learner can guess or estimate the step size, which depends only on the scalar quantity $\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}}$. In the next section, we will consider the case where even that information is unknown.

The problem in this scenario is that without prior knowledge of the second moments of the attributes, the learner cannot calculate the optimal q_i -s via equation (4). To address this issue we split the learning into two phases: In the first phase we run on the first m_1 training examples and estimate the second moments by sampling the attributes uniformly at random. In the second phase we run on the next m_2 training examples, and perform the regular DDAERR algorithm, with a slight modification - in the calculation of the q_i -s, we use an upper confidence interval instead of the second moments themselves. We assume m_2 is on the order of m . This approach is the basis for our Two-Phased DDAERR algorithm (Algorithm 2). The estimate for $\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}}$ does not assist in the calculation of the q_i -s, but will give us the optimal step size, η .

Note that in practice, one can actually run the AERR algorithm during the first phase, in order to obtain a better starting point for the second phase. We ignore this improvement in our analysis below, but incorporate it in the experiments presented in section 5.

There are other variants of this type, the most apparent of them is to use the same samples that estimate the gradient to estimate the moments themselves. This method, even though in some cases may be superior to our method, will not yield better results in the worst case scenarios because we may never get accurate enough estimations for some of the attributes.

The expected risk bound of the algorithm is formulated in the following theorem.

Theorem 3.6. *Assume the distribution \mathcal{D} is such that $\|\mathbf{x}\|_2 \leq 1$ and $|y| \leq B$ with probability 1. Assume further that the value $\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}}$ is known. Let $\bar{\mathbf{w}}$ be the output of Two-Phased DDAERR*

³If $\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}} = d$ it is easy to see that $\mathbb{E}_D[x_i] = \frac{1}{d}$ for all $i \in [d]$. In this case, all the q_i -s are equal to $\frac{1}{d}$ and the DDAERR and AERR algorithms coincide.

Algorithm 2 Two-Phased DDAERR

Parameters: $m_1, m_2, \delta, B, \eta > 0$

Input: training set $S = \{(\mathbf{x}_t, y_t)\}_{t \in [m_1 + m_2]}$ and $k > 0$ **Output:** regressor $\bar{\mathbf{w}}$ with $\|\bar{\mathbf{w}}\|_2 \leq B$

- 1: Initialize $\mathbf{w}_1 \neq 0$, $\|\mathbf{w}_1\|_2 \leq B$ arbitrarily
 - 2: Initialize \mathbf{A} , *counts* and *square_sums* - arrays of size d with zeros
 - 3: **for** $t = 1$ to m_1 **do**
 - 4: **for** $r = 1$ to $k + 1$ **do**
 - 5: Pick $i_{t,r} \in [d]$ uniformly at random
 - 6: $\text{counts}[i_{t,r}] \leftarrow \text{counts}[i_{t,r}] + 1$
 - 7: $\text{square_sums}[i_{t,r}] \leftarrow \text{square_sums}[i_{t,r}] + \mathbf{x}_t[i_{t,r}]^2$
 - 8: $\mathbf{A}[i_{t,r}] \leftarrow \frac{\text{square_sums}[i_{t,r}]}{\text{counts}[i_{t,r}]}$
 - 9: **end for**
 - 10: **end for**
 - 11: $\epsilon \leftarrow \frac{d \log \frac{2d}{\delta}}{(k+1)m_1}$
 - 12: Run GAERR with $q_i = \frac{\sqrt{\mathbf{A}[i] + \frac{13}{6}\epsilon}}{\sum_{j=1}^d \sqrt{\mathbf{A}[j] + \frac{13}{6}\epsilon}}$ on the following m_2 examples and return its output
-

when run with

$$\eta = \max \left(\sqrt{\frac{k}{6dm_2}}, \sqrt{\frac{k}{m_2 \left(2 \|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}} + 2\sqrt{\frac{5}{3}}d \sqrt{\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}}} \sqrt{\frac{d \log \frac{2d}{\delta}}{(k+1)m_1}} + k \right)}} \right).$$

Then for all m_1 and for any $\mathbf{w}^* \in \mathbb{R}^d$ with $\|\mathbf{w}^*\|_2 \leq B$, with probability 1 over the first phase, we have

$$\mathbb{E}_{D, A_2} [L_{\mathcal{D}}(\bar{\mathbf{w}})] - L_{\mathcal{D}}(\mathbf{w}^*) \leq \frac{4B^2}{\sqrt{m_2}} \sqrt{\frac{6d}{k}}.$$

Also, with probability $\geq 1 - \delta$ over the first phase, we have

$$\mathbb{E}_{D, A_2} [L_{\mathcal{D}}(\bar{\mathbf{w}})] - L_{\mathcal{D}}(\mathbf{w}^*) \leq \frac{4B^2}{\sqrt{m_2}} \sqrt{\frac{2}{k} \|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}} + \frac{2}{k} \sqrt{\frac{5}{3}}d \sqrt{\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}}} \sqrt{\frac{d \log \frac{2d}{\delta}}{(k+1)m_1}} + 1}.$$

With probability 1 over the first phase, regardless of the value of m_1 , the expected risk bound is at most $O\left(\frac{B^2}{\sqrt{km_2}}\sqrt{d}\right)$, which is the same bound of the AERR algorithm. This means that the Two-Phased DDAERR algorithm performs with probability 1 over the first phase as well as the AERR algorithm, up to a constant factor. Second, as m_1 increases, the expected risk bound turns to $O\left(\frac{B^2}{\sqrt{km_2}}\sqrt{\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}} + d\sqrt{\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}}} \sqrt{\frac{d \log \frac{2d}{\delta}}{(k+1)m_1}} + k}\right)$. Therefore, if $m_1 \gg \frac{d\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}} \log \frac{2d}{\delta}}{k+1}$,

we achieve an improvement over the AERR algorithm. If $m_1 \geq \frac{d^3 \log \frac{2d}{\delta}}{(k+1)\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}}}$, the bound becomes

$O\left(\frac{B^2}{\sqrt{km_2}}\sqrt{\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}} + k}\right)$, which is the same bound as in the regular DDAERR algorithm which assumes prior knowledge of the second moment of the attributes.

The conclusion is that even if we do not have prior knowledge of the second moments of the attributes, but can guess $\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}}$, we still should prefer our Two-Phased DDAERR algorithm over the AERR algorithm. In the next section, we analyze the case where even $\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}}$ is unknown.

3.2.1 Proof of Theorem 3.6

The main goal of the proof is to bound the expected squared 2-norm of the gradient estimator from above. By using Lemma 3.4, all that remains is to upper bound $\mathbb{E}_{D,A_2}[\|\tilde{\mathbf{x}}_{t,r}\|_2^2]$. In the next lemma we show two different upper bounds on $\mathbb{E}_{D,A_2}[\|\tilde{\mathbf{x}}_{t,r}\|_2^2]$. The first states that with probability 1 over the first phase $\mathbb{E}_{D,A_2}[\|\tilde{\mathbf{x}}_{t,r}\|_2^2] \leq 5d$, meaning that up to a constant factor the bound is the same as in the AERR algorithm. The second bound decreases in ϵ , and will help up to analyze the convergence rate of the algorithm.

Lemma 3.7. *For all m_1 and $t > m_1$, with probability 1 over the first phase, we have*

$$\mathbb{E}_{D,A_2}[\|\tilde{\mathbf{x}}_{t,r}\|_2^2] \leq 5d,$$

and with probability $\geq 1 - \delta$ over the first phase, we have

$$\mathbb{E}_{D,A_2}[\|\tilde{\mathbf{x}}_{t,r}\|_2^2] \leq 2\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}} + 2\sqrt{\frac{5}{3}}d\sqrt{\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}}}\sqrt{\epsilon}.$$

The proof can be found in Appendix B.4.

We will treat each bound separately, and later join the results into a single lemma. First, we prove that even if we do not have an estimate for $\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}}$, with a proper choice of η , our Two-Phased DDAERR algorithm still performs with probability 1 over the first phase as well as the AERR algorithm, up to a constant factor.

Lemma 3.8. *Let $\bar{\mathbf{w}}$ be the output of Two-Phased DDAERR when run with $\eta = \sqrt{\frac{k}{6dm_2}}$. Then with probability 1 over the first phase, we have for all m_1 and for any $\mathbf{w}^* \in \mathbb{R}^d$ with $\|\mathbf{w}^*\|_2 \leq B$,*

$$\mathbb{E}_{D,A_2}[L_{\mathcal{D}}(\bar{\mathbf{w}})] - L_{\mathcal{D}}(\mathbf{w}^*) \leq 4B^2\sqrt{\frac{6d}{km_2}}.$$

The proof can be found in appendix B.5.

Now, if we do have an estimate $H \geq \|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}}$, we can use it to calculate an appropriate step size and to bound the risk, as shown in the next lemma.

Lemma 3.9. *Assume we have a value H that satisfies $H \geq \|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}}$. Let $\bar{\mathbf{w}}$ be the output of Two-Phased DDAERR when run with $\eta = \frac{1}{\sqrt{m_2(\frac{2}{k}H + \frac{2}{k}\sqrt{\frac{5}{3}}d\sqrt{H}\sqrt{\epsilon} + 1)}}$. Then with probability $\geq 1 - \delta$ over the first phase, we have for all m_1 and for any $\mathbf{w}^* \in \mathbb{R}^d$ with $\|\mathbf{w}^*\|_2 \leq B$,*

$$\mathbb{E}_{D,A_2}[L_{\mathcal{D}}(\bar{\mathbf{w}})] - L_{\mathcal{D}}(\mathbf{w}^*) \leq \frac{4B^2}{\sqrt{m_2}}\sqrt{\frac{2}{k}H + \frac{2}{k}\sqrt{\frac{5}{3}}d\sqrt{H}\sqrt{\epsilon} + 1}.$$

The proof can be found in appendix B.6.

This lemma gives a non-trivial expected risk bound only if ϵ is small enough, but when m_1 is small, this is not necessarily the case. Therefore, we would like to unite these two lemmas to ensure that even in the worst case, we won't have a worse bound than the AERR algorithm.

Lemma 3.10. *Assume we know a value H that satisfies $H \geq \|\mathbb{E}_D [\mathbf{x}^2]\|_{\frac{1}{2}}$. Let $\bar{\mathbf{w}}$ be the output of Two-Phased DDAERR when run with*

$$\eta = \max \left(\sqrt{\frac{k}{6dm_2}}, \sqrt{\frac{k}{m_2 \left(2H + 2\sqrt{\frac{5}{3}}d\sqrt{H}\sqrt{\frac{d \log \frac{2d}{\delta}}{(k+1)m_1}} + k \right)}} \right).$$

Then for all m_1 and for any $\mathbf{w}^ \in \mathbb{R}^d$ with $\|\mathbf{w}^*\|_2 \leq B$, with probability 1 over the first phase, we have*

$$\mathbb{E}_{D, A_2} [L_{\mathcal{D}}(\bar{\mathbf{w}})] - L_{\mathcal{D}}(\mathbf{w}^*) \leq \frac{4B^2}{\sqrt{m_2}} \sqrt{\frac{6d}{k}}.$$

Also, with probability $\geq 1 - \delta$ over the first phase, we have

$$\mathbb{E}_{D, A_2} [L_{\mathcal{D}}(\bar{\mathbf{w}})] - L_{\mathcal{D}}(\mathbf{w}^*) \leq \frac{4B^2}{\sqrt{m_2}} \sqrt{\frac{2}{k}H + \frac{2}{k}\sqrt{\frac{5}{3}}d\sqrt{H}\sqrt{\frac{d \log \frac{2d}{\delta}}{(k+1)m_1}} + 1}.$$

The proof can be found in appendix B.7.

We could always naively bound $\|\mathbb{E}_D [\mathbf{x}^2]\|_{\frac{1}{2}}$ by d , but then, even if m_1 tends to infinity, the bound will not be better than the bound of the AERR algorithm. However, if we do have prior knowledge upon the value $\|\mathbb{E}_D [\mathbf{x}^2]\|_{\frac{1}{2}}$, it is straightforward to use this lemma to prove Theorem 3.6.

3.3 Unknown Second Moment Scenario

In this section, we analyze the case in which we do not have prior knowledge of the second moments of the attributes and on the value of $\|\mathbb{E}_D [\mathbf{x}^2]\|_{\frac{1}{2}}$. This scenario may accrue if the learner is faced with a new learning task, and knows nothing about the distribution of the attributes. The problem here, besides not being able to calculate the optimal q_i -s, is that we also cannot calculate the optimal step size, η .

Our solution to this scenario is to use again the Two-Phased DDAERR algorithm (Algorithm 2), and calculate an accurate enough estimation of $\|\mathbb{E}_D [\mathbf{x}^2]\|_{\frac{1}{2}}$.

Lemma 3.11. *We can estimate $\|\mathbb{E}_D [\mathbf{x}^2]\|_{\frac{1}{2}}$ by the estimator $H = \|2\mathbf{A} + \frac{10}{3}\epsilon\|_{\frac{1}{2}}$, which satisfies with probability $\geq 1 - \delta$, $\|\mathbb{E}_D [\mathbf{x}^2]\|_{\frac{1}{2}} \leq \|2\mathbf{A} + \frac{10}{3}\epsilon\|_{\frac{1}{2}} \leq 8\|\mathbb{E}_D [\mathbf{x}^2]\|_{\frac{1}{2}} + \frac{34}{3}d^2\epsilon$.*

Proof. First, using the second inequality in equation (15) we have with probability $\geq 1 - \delta$, that $\|\mathbb{E}_D [\mathbf{x}^2]\|_{\frac{1}{2}} \leq \|2\mathbf{A} + \frac{10}{3}\epsilon\|_{\frac{1}{2}}$. Using the first inequality in equation (15) and the identity $\|\mathbf{a} + \mathbf{b}\|_{\frac{1}{2}} \leq 2\|\mathbf{a}\|_{\frac{1}{2}} + 2\|\mathbf{b}\|_{\frac{1}{2}}$ we can see that with probability $\geq 1 - \delta$,

$$\left\| 2\mathbf{A} + \frac{10}{3}\epsilon \right\|_{\frac{1}{2}} \leq \left\| 4\mathbb{E}_D [\mathbf{x}^2] + \frac{14}{6}\epsilon + \frac{10}{3}\epsilon \right\|_{\frac{1}{2}} \leq 8\|\mathbb{E}_D [\mathbf{x}^2]\|_{\frac{1}{2}} + \frac{34}{3}d^2\epsilon. \quad \square \quad (6)$$

Using this estimate we can prove our main theorem of this section.

Theorem 3.12. *Assume the distribution \mathcal{D} is such that $\|\mathbf{x}\|_2 \leq 1$ and $|y| \leq B$ with probability 1. Let $\bar{\mathbf{w}}$ be the output of Two-Phased DDAERR when run with*

$$\eta = \max \left(\sqrt{\frac{k}{6dm_2}}, \sqrt{\frac{k}{m_2 \left(2 \left\| 2\mathbf{A} + \frac{10}{3}\epsilon \right\|_{\frac{1}{2}} + 2\sqrt{\frac{5}{3}}d \sqrt{\left\| 2\mathbf{A} + \frac{10}{3}\epsilon \right\|_{\frac{1}{2}}} \sqrt{\frac{d \log \frac{2d}{\delta}}{(k+1)m_1}} + k \right)}} \right).$$

Then for all m_1 and for any $\mathbf{w}^* \in \mathbb{R}^d$ with $\|\mathbf{w}^*\|_2 \leq B$, with probability 1 over the first phase, we have

$$\mathbb{E}_{D,A_2} [L_{\mathcal{D}}(\bar{\mathbf{w}})] - L_{\mathcal{D}}(\mathbf{w}^*) \leq \frac{4B^2}{\sqrt{m_2}} \sqrt{\frac{6d}{k}}.$$

Also, with probability $\geq 1 - \delta$ over the first phase, we have

$$\mathbb{E}_{D,A_2} [L_{\mathcal{D}}(\bar{\mathbf{w}})] - L_{\mathcal{D}}(\mathbf{w}^*) \leq \frac{16B^2}{\sqrt{m_2}} \sqrt{\frac{1}{k} \left(\sqrt{\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}}} + d \sqrt{\frac{2d \log \frac{2d}{\delta}}{(k+1)m_1}} \right)^2} + 1.$$

If we examine the bound we can see that with probability $\geq 1 - \delta$ over the first phase, regardless of the value of m_1 , the expected risk bound is at most $O\left(\frac{B^2}{\sqrt{km_2}}\sqrt{d}\right)$, which is the same bound of the AERR algorithm. This means that the Two-Phased DDAERR algorithm performs with high probability over the first phase as well as the AERR algorithm, up to a constant factor. Second, as

m_1 increases, the expected risk bound turns to $O\left(\frac{B^2}{\sqrt{km_2}} \sqrt{\left(\sqrt{\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}}} + d \sqrt{\frac{2d \log \frac{2d}{\delta}}{(k+1)m_1}}\right)^2 + k}\right)$.

Therefore, if $m_1 \gg \frac{d^2 \log \frac{2d}{\delta}}{k+1}$, we achieve an improvement over the AERR algorithm. If $m_1 \geq \frac{d^3 \log \frac{2d}{\delta}}{(k+1)\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}}}$, the bound becomes $O\left(\frac{B^2}{\sqrt{km_2}} \sqrt{\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}} + k}\right)$, which is the same bound as in the regular DDAERR algorithm with prior knowledge of the second moment of the attributes.

The conclusion is that even if we do not have prior knowledge of the second moments of the attributes and on $\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}}$, we still should prefer our Two-Phased DDAERR algorithm over the AERR algorithm.

It is interesting to compare between the known $\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}}$ case and the unknown $\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}}$ case. Even though the sampling probabilities are the same, in the unknown $\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}}$ case we need $\frac{d}{\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}}}$ more samples to reach the regime where our algorithm significantly improves on AERR. The reason for this is that the expected risk bound is highly dependent on the choice of the step size η , and calculating the optimal value requires knowledge of $\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}}$ which is harder to estimate than the moments themselves, as the estimation errors of attributes build up.

3.3.1 Proof of Theorem 3.12

The proof is straightforward using Lemma 3.10. First, by denoting $H = \|2\mathbf{A} + \frac{10}{3}\epsilon\|_{\frac{1}{2}}$, and using

$$\eta = \frac{1}{\sqrt{m_2 \left(\frac{2}{k}H + \frac{2}{k}\sqrt{\frac{5}{3}}d\sqrt{H}\sqrt{\epsilon} + 1 \right)}} = \frac{1}{\sqrt{m_2 \left(\frac{2}{k}\|2\mathbf{A} + \frac{10}{3}\epsilon\|_{\frac{1}{2}} + \frac{2}{k}\sqrt{\frac{5}{3}}d\sqrt{\|2\mathbf{A} + \frac{10}{3}\epsilon\|_{\frac{1}{2}}}\sqrt{\epsilon} + 1 \right)}}$$

We can see that

$$\begin{aligned} & \frac{4B^2}{\sqrt{m_2}} \sqrt{\frac{2}{k}H + \frac{2}{k}\sqrt{\frac{5}{3}}d\sqrt{H}\sqrt{\epsilon} + 1} \\ & \leq \frac{4B^2}{\sqrt{m_2}} \sqrt{\frac{2}{k}\|2\mathbf{A} + \frac{10}{3}\epsilon\|_{\frac{1}{2}} + \frac{2}{k}\sqrt{\frac{5}{3}}d\sqrt{\|2\mathbf{A} + \frac{10}{3}\epsilon\|_{\frac{1}{2}}}\sqrt{\epsilon} + 1} \\ & \leq \frac{4B^2}{\sqrt{m_2}} \sqrt{\frac{2}{k} \left(8\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}} + \frac{34}{3}d^2\epsilon \right) + \frac{2}{k}\sqrt{\frac{5}{3}}d\sqrt{\left(8\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}} + \frac{34}{3}d^2\epsilon \right)}\sqrt{\epsilon} + 1} \\ & \leq \frac{4B^2}{\sqrt{m_2}} \sqrt{\frac{16}{k}\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}} + \frac{68}{3k}d^2\epsilon + \frac{2}{k}\sqrt{\frac{40}{3}}d\sqrt{\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}}}\sqrt{\epsilon} + \frac{2}{k}\sqrt{\frac{170}{9}}d^2\epsilon + 1} \\ & \leq \frac{4B^2}{\sqrt{m_2}} \sqrt{\frac{1}{k} \left(16\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}} + 2\sqrt{\frac{40}{3}}d\sqrt{\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}}}\sqrt{\epsilon} + 32d^2\epsilon \right) + 1} \\ & \leq \frac{16B^2}{\sqrt{m_2}} \sqrt{\frac{1}{k} \left(\sqrt{\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}}} + d\sqrt{2\epsilon} \right)^2 + 1}. \end{aligned}$$

Using Lemma 3.10 and plugging in $\epsilon = \frac{d \log \frac{2d}{\delta}}{(k+1)m_1}$ finishes the proof.

4 Attribute Efficient Lasso Regression

In this section we present our algorithms for lasso regression, where the loss is again the squared loss, $\ell(\mathbf{w}; \mathbf{x}_t, y_t) = \frac{1}{2}(\langle \mathbf{w}, \mathbf{x}_t \rangle - y_t)^2$, and the 1-norm bound is $\|\mathbf{w}\|_1 \leq B$. The generic approach to the lasso attribute efficient scenario, which we call the General Attribute Efficient Lasso Regression (GAELR) algorithm and is presented in Algorithm 3, was first developed in [4] and is based on a stochastic variant of the Exponentiated Gradient (EG) algorithm with gradient estimates, developed in [8].

The EG algorithm goes over the training set, and for each example builds an unbiased estimator of the gradient and clips it (where the *clip* operation is defined as $\text{clip}(x, c) = \max\{\min\{x, c\}, -c\}$) to make the updates more robust. Afterwards, the algorithm updates \mathbf{w}_t by performing multiplicative updates of size η . The result is projected over the L_1 ball of size B , yielding \mathbf{w}_{t+1} . At the end, the algorithm outputs the average of all \mathbf{w}_t . The algorithm converges to the global minimum, as the minimization problem is convex in \mathbf{w} .

The GAELR algorithm build the unbiased gradient estimates similarly to the GAERR algorithm, with a slight modification: When estimating the inner product, instead of sampling one

sample with probability $p_{j_t} = \frac{w_{t,j_t}^2}{\|\mathbf{w}_t\|_2^2}$, it samples it with probability $p_{j_t} = \frac{|\mathbf{w}_t[j_t]|}{\|\mathbf{w}_t\|_1}$, as the lasso scenario has a bound on the 1-norm of the predictor.

Algorithm 3 GAELR

Parameters: $B, \eta > 0$ and q_i for $i \in [d]$

Input: training set $S = \{(\mathbf{x}_t, y_t)\}_{t \in [m]}$ and $k > 0$

Output: regressor $\bar{\mathbf{w}}$ with $\|\bar{\mathbf{w}}\|_1 \leq B$

```

1: Initialize  $\mathbf{z}_1^+ \leftarrow \mathbf{1}_d, \mathbf{z}_1^- \leftarrow \mathbf{1}_d$ 
2: for  $t = 1$  to  $m$  do
3:    $\mathbf{w}_t \leftarrow (\mathbf{z}_t^+ - \mathbf{z}_t^-) B / (\|\mathbf{z}_t^+\|_1 + \|\mathbf{z}_t^-\|_1)$ 
4:   for  $r = 1$  to  $k$  do
5:     Pick  $i_{t,r} \in [d]$  with probability  $q_{i_{t,r}}$  and observe  $\mathbf{x}_t[i_{t,r}]$ 
6:      $\tilde{\mathbf{x}}_{t,r} \leftarrow \frac{1}{q_{i_{t,r}}} \mathbf{x}_t[i_{t,r}] \cdot \mathbf{e}_{i_{t,r}}$ 
7:   end for
8:    $\tilde{\mathbf{x}}_t \leftarrow \frac{1}{k} \sum_{r=1}^k \tilde{\mathbf{x}}_{t,r}$ 
9:   Choose  $j_t \in [d]$  with probability  $p_{j_t} = \frac{|\mathbf{w}_t[j_t]|}{\|\mathbf{w}_t\|_1}$  and observe  $\mathbf{x}_t[j_t]$ 
10:   $\tilde{\phi}_t \leftarrow \frac{w_{t,j_t}}{p_{j_t}} \mathbf{x}_t[j_t] - y_t$ 
11:   $\tilde{\mathbf{g}}_t \leftarrow \tilde{\phi}_t \cdot \tilde{\mathbf{x}}_t$ 
12:  for  $i = 1$  to  $d$  do
13:     $\bar{\mathbf{g}}_t[i] = \text{clip}(\tilde{\mathbf{g}}_t[i], 1/\eta)$ 
14:     $\mathbf{z}_{t+1}^+[i] \leftarrow \mathbf{z}_t^+[i] \cdot \exp(-\eta \bar{\mathbf{g}}_t[i])$ 
15:     $\mathbf{z}_{t+1}^-[i] \leftarrow \mathbf{z}_t^-[i] \cdot \exp(+\eta \bar{\mathbf{g}}_t[i])$ 
16:  end for
17: end for
18:  $\bar{\mathbf{w}} \leftarrow \frac{1}{m} \sum_{t=1}^m \mathbf{w}_t$ 

```

The expected risk bound of the GAELR algorithm is presented in the next theorem which is a slightly more general version of Theorem 3.4 in [4].

Theorem 4.1. *Assume the distribution \mathcal{D} is such that $\|\mathbf{x}\|_\infty \leq 1$ and $|y| < B$ with probability 1. Let $\bar{\mathbf{w}}$ be the output of GAELR, when run with step size $\eta \leq \frac{1}{2G}$ where $\max_t \left\| \mathbb{E}_{D,A} [\tilde{\mathbf{g}}_t^2] \right\|_\infty \leq G^2$. Then for any $\mathbf{w}^* \in \mathbb{R}^d$ with $\|\mathbf{w}^*\|_1 \leq B$,*

$$\mathbb{E}_{D,A} [L_{\mathcal{D}}(\bar{\mathbf{w}})] \leq L_{\mathcal{D}}(\mathbf{w}^*) + B \left(\frac{\log 2d}{\eta m} + 5\eta G^2 \right).$$

The general idea of the proof is that $\tilde{\mathbf{g}}_t$ is an unbiased estimator of the gradient, therefore we can use the standard analysis of the EG algorithm. The full proof can be found in appendix B.8.

The AELR algorithm is one variant of the GAELR algorithm. It was presented in [4] and uses uniform sampling to estimate \mathbf{x}_t . In our GAELR notation it uses

$$q_i = \frac{1}{d} \quad \forall i \in [d].$$

The authors prove (Lemma 3.8 in [4]) that for the AELR algorithm, $G^2 \leq 8B^2 d/k$, which together with Theorem 4.1 and using $\eta = \frac{2B}{G\sqrt{m}}$ yields an expected risk bound of $4B^2 \sqrt{\frac{10d \log 2d}{km}}$.

Similarly to the ridge scenario, by analyzing the bound, we show that we can improve the bound in a data-dependent manner: Theorem 4.1 tells us that the expected risk bound is proportional to G , therefore we wish to develop a sampling method that minimizes the infinity norm of the gradient estimator.

The gradient estimate consist of estimating the inner product and estimating \mathbf{x}_t . The next lemma will assist in bounding the infinity norm of $\tilde{\mathbf{x}}_t^2$.

Lemma 4.2. *For every distribution (q_1, \dots, q_d) where $q_i \geq 0$ and $i \in [d]$, we have $\|\mathbb{E}_{D,A} [\tilde{\mathbf{x}}_t^2]\|_\infty = \max_i \frac{1}{k} \mathbb{E}_{D,A} [\tilde{\mathbf{x}}_{t,r}^2 [i]] + \frac{k-1}{k} \mathbb{E}_D [\|\mathbf{x}\|_\infty]^2$.*

The proof can be found in appendix B.9.

Since

$$\mathbb{E}_{D,A} [\tilde{\mathbf{x}}_{t,r}^2 [i]] = \frac{1}{q_i} \mathbb{E}_D [x_i^2], \quad (7)$$

in order to minimize the infinity norm of the estimator, we need to solve the following optimization problem:

$$\begin{aligned} & \underset{q_i}{\text{minimize}} && \max_i \frac{1}{k q_i} \mathbb{E}_D [x_i^2] + \frac{k-1}{k} \mathbb{E}_D [\|\mathbf{x}\|_\infty]^2 \\ & \text{subject to} && \sum_{i=1}^d q_i = 1, \forall i \ q_i \geq 0. \end{aligned}$$

This problem is equivalent to

$$\begin{aligned} & \underset{q_i}{\text{minimize}} && \max_i \frac{1}{q_i} \mathbb{E}_D [x_i^2] \\ & \text{subject to} && \sum_{i=1}^d q_i = 1, \forall i \ q_i \geq 0. \end{aligned} \quad (8)$$

The next lemma gives the optimal value of the q_i -s.

Lemma 4.3. *The solution to the optimization problem defined in (8) is $q_i = \frac{\mathbb{E}_D [x_i^2]}{\sum_{j=1}^d \mathbb{E}_D [x_j^2]}$.*

The proof can be found in appendix B.10.

The next lemma will assist in bounding the square of the estimator of the inner product (minus the label).

Lemma 4.4. *Using our sampling method we have $\mathbb{E}_{D,A} [\tilde{\phi}_t^2] \leq 4B^2$.*

The proof can be found in appendix B.11.

As in the ridge scenario, we could have tried to optimized the sampling probabilities of the inner product estimation. However, since $\mathbb{E}_{D,A} [\tilde{\phi}_t^2]$ is calculated using the same method as in the ridge scenario, the optimal sampling probabilities remain $p_i = \frac{\sqrt{\mathbf{w}_{t,i}^2 \mathbb{E}_D [x_i^2]}}{\sum_{j=1}^d \sqrt{\mathbf{w}_{t,j}^2 \mathbb{E}_D [x_j^2]}}$, but we will still ignore this improvement in our analysis.

Altogether, we can formulate a lemma that will bound the gradient estimate.

Lemma 4.5. *The GAELR algorithm generates gradient estimates that for all t , $\left\| \mathbb{E}_{D,A} [\tilde{\mathbf{g}}_t^2] \right\|_\infty \leq 4B^2 \left(\frac{1}{k} \left\| \mathbb{E}_{D,A} [\tilde{\mathbf{x}}_{t,r}^2] \right\|_\infty + 1 \right)$.*

Proof. This lemma follows directly from Lemmas 4.2 and 4.4, using the independence of $\tilde{\mathbf{x}}_t$ and $\tilde{\phi}_t$ given \mathbf{x}_t and $\|\mathbf{x}\|_\infty \leq 1$. \square

4.1 Known Second Moment Scenario

If we assume we have prior knowledge of the second moment of each attribute, namely $\mathbb{E}_D [x_i^2]$ for all $i \in [d]$, we can use Lemma 4.3 to calculate the optimal values of the q_i -s. This is the idea behind our DDAELR (Data-Dependent Attribute Efficient Lasso Regression) algorithm.

The expected risk bound of the algorithm is formulated in the next theorem.

Theorem 4.6. *Assume the distribution \mathcal{D} is such that $\|\mathbf{x}\|_\infty \leq 1$ and $|y| \leq B$ with probability 1 and $\mathbb{E}_D [x_i^2]$ are known for $i \in [d]$. Let $\bar{\mathbf{w}}$ be the output of DDAELR, when run with $\eta = \frac{1}{2B} \sqrt{\frac{\log 2d}{5m(\frac{1}{k} \|\mathbb{E}_D [\mathbf{x}^2]\|_1 + 1)}}$. If $m \geq \log 2d$ then for any $\mathbf{w}^* \in \mathbb{R}^d$ with $\|\mathbf{w}^*\|_1 \leq B$,*

$$\mathbb{E}_{D,A} [L_{\mathcal{D}}(\bar{\mathbf{w}})] \leq L_{\mathcal{D}}(\mathbf{w}^*) + 4B^2 \sqrt{\frac{5 \log 2d \left(\frac{1}{k} \|\mathbb{E}_D [\mathbf{x}^2]\|_1 + 1 \right)}{m}}.$$

Proof of Theorem 4.6. If $m \geq \log 2d$, we have $\eta \leq \frac{1}{2G}$ and the theorem follows directly from Theorem 4.1, Lemma 4.5, equation (7) and the calculated q_i -s in Lemma 4.3. \square

Recalling that with probability 1 we have $\|\mathbf{x}\|_\infty \leq 1$, it is easy to see that $\|\mathbb{E}_D [\mathbf{x}^2]\|_1 \leq d$, therefore the DDAELR algorithm always performs at least as well as the AELR algorithm⁴. However, $\|\mathbb{E}_D [\mathbf{x}^2]\|_1$ may also be much smaller than d , in cases where the second moments varies between attributes or the vector is sparse. In these cases, we may gain a significant improvement. For example, if we consider a harmonic attribute decay such as $\mathbb{E}_D [x_i^2] = \frac{1}{i}$, we have $\|\mathbb{E}_D [\mathbf{x}^2]\|_1 = O(\log d)$ which is significantly smaller than d .

4.2 Unknown Second Moment Scenario

The solution presented in the previous section requires exact knowledge of $\mathbb{E}_D [x_i^2]$ for all i , which may not be available when the learner is faced with a new learning task. Thus, we turn to consider the case where the moments are initially unknown.

We take a similar approach to the Two-Phased DDAERR algorithm: in the first phase, we estimate the second moments by uniform sampling, exactly as in the Two-Phased DDAERR algorithm. In the second phase, we run the DAELR with modified q_i s which use an upper confidence interval instead of the second moments themselves. This approach is the basis for our Two-Phased DDAELR algorithm (Algorithm 4).

As in the Two-Phased DDAERR algorithm, during the first phase one can actually run the AELR algorithm in order to obtain a better starting point for the second phase, but we will ignore this improvement in our analysis.

The expected risk bound of the algorithm is formulated in the following theorem.

⁴If $\|\mathbb{E}_D [\mathbf{x}^2]\|_1 = d$ it is easy to see that $\mathbb{E}_D [x_i] = 1$ for all $i \in [d]$. In this case, all the q_i -s are equal to $\frac{1}{d}$ and the DDAELR and AELR algorithms coincide.

Algorithm 4 Two-Phased DDAELR

Parameters: $m_1, m_2, \delta, B, \eta > 0$

Input: training set $S = \{(\mathbf{x}_t, y_t)\}_{t \in [m_1 + m_2]}$ and $k > 0$ **Output:** regressor $\bar{\mathbf{w}}$ with $\|\bar{\mathbf{w}}\| \leq B$

- 1: Initialize $\mathbf{w}_1 \neq \mathbf{0}$, $\|\mathbf{w}_1\|_2 \leq B$ arbitrarily
 - 2: Initialize \mathbf{A} , *counts* and *square_sums* - arrays of size d with zeros
 - 3: **for** $t = 1$ to m_1 **do**
 - 4: **for** $r = 1$ to $k + 1$ **do**
 - 5: Pick $i_{t,r} \in [d]$ uniformly at random
 - 6: $\text{counts}[i_{t,r}] \leftarrow \text{counts}[i_{t,r}] + 1$
 - 7: $\text{square_sums}[i_{t,r}] \leftarrow \text{square_sums}[i_{t,r}] + \mathbf{x}_t[i_{t,r}]^2$
 - 8: $\mathbf{A}[i_{t,r}] \leftarrow \frac{\text{square_sums}[i_{t,r}]}{\text{counts}[i_{t,r}]}$
 - 9: **end for**
 - 10: **end for**
 - 11: $\epsilon \leftarrow \min\left(\frac{d \log \frac{2d}{\delta}}{(k+1)m_1}, 1\right)$
 - 12: Run GAELR with $q_i = \frac{\mathbf{A}[i] + \frac{13}{6}\epsilon}{\sum_{j=1}^d (\mathbf{A}[j] + \frac{13}{6}\epsilon)}$ on the following m_2 examples and return its output
-

Theorem 4.7. Assume the distribution \mathcal{D} is such that $\|\mathbf{x}\|_\infty \leq 1$ and $|y| \leq B$ with probability

1. Let $\bar{\mathbf{w}}$ be the output of DDAELR, when run with $\eta = \sqrt{\frac{k \log 2d}{20B^2m_2 \left(8\|\mathbf{A}\|_1 + 20d \min\left(\frac{d \log \frac{2d}{\delta}}{(k+1)m_1}, 1\right) + k\right)}}$.

If $m_2 \geq \log 2d$ then for any m_1 and for any $\mathbf{w}^* \in \mathbb{R}^d$ with $\|\mathbf{w}^*\|_1 \leq B$, with probability 1 over the first phase we have,

$$\mathbb{E}_{D, A_2} [L_{\mathcal{D}}(\bar{\mathbf{w}})] - L_{\mathcal{D}}(\mathbf{w}^*) \leq 61B^2 \sqrt{\frac{d \log 2d}{km_2}}.$$

Also, with probability $1 - \delta$ over the first phase we have

$$\mathbb{E}_{D, A_2} [L_{\mathcal{D}}(\bar{\mathbf{w}})] - L_{\mathcal{D}}(\mathbf{w}^*) \leq 4B^2 \sqrt{\frac{5 \left(16 \|\mathbb{E}_D[\mathbf{x}^2]\|_1 + \frac{88d}{3} \min\left(\frac{d \log \frac{2d}{\delta}}{(k+1)m_1}, 1\right) + k\right) \log 2d}{km_2}}.$$

With probability 1 over the first phase, regardless of the value of m_1 , the expected risk bound is at most $O\left(\frac{B^2}{\sqrt{km_2}} \sqrt{d \log d}\right)$, which is the same bound of the AELR algorithm. This means that the Two-Phased DDAELR algorithm performs with probability 1 over the first phase as well as the AELR algorithm, up to a constant factor. Second, as m_1 increases, the expected risk bound becomes $O\left(\frac{B^2}{\sqrt{km_2}} \sqrt{\left(\|\mathbb{E}_D[\mathbf{x}^2]\|_1 + \frac{d^2 \log \frac{2d}{\delta}}{(k+1)m_1} + k\right) \log d}\right)$. Therefore, if $m_1 \gg \frac{d \log \frac{2d}{\delta}}{k+1}$, we achieve an

improvement over the AELR algorithm. If $m_1 \geq \frac{d^2 \log \frac{2d}{\delta}}{(k+1)\|\mathbb{E}_D[\mathbf{x}^2]\|_1}$, the expected risk bound turns to $O\left(\frac{B^2}{\sqrt{km_2}} \sqrt{\|\mathbb{E}_D[\mathbf{x}^2]\|_1 + k}\right)$, which is the same bound as in the regular DDAELR algorithm with prior knowledge of the second moment of the attributes.

The conclusion is that even if we do not have prior knowledge of the second moments of the attributes, we still should prefer our Two-Phased DDAELR algorithm over the AELR algorithm.

It is also interesting to compare these improvement regimes to those of the Two-Phased DDAERR algorithm. If we analogize $\|\mathbb{E}_D [\mathbf{x}^2]\|_{\frac{1}{2}}$ to $\|\mathbb{E}_D [\mathbf{x}^2]\|_1$, the regimes for the lasso scenario are better by a factor d than the corresponding regimes for the ridge scenario. The reason for this is that for any i , $\mathbb{E}_D [x_i^2]$ is much easier to estimate by sampling than $\sqrt{\mathbb{E}_D [x_i^2]}$, because the square root is not a Lipschitz function.

4.2.1 Proof of Theorem 4.7

The main goal of the proof is to bound the expected squared infinity-norm of the gradient estimator from above. By using Lemma 4.5, all that remains is to upper bound $\|\mathbb{E}_{D,A} [\tilde{\mathbf{x}}_{t,r}^2]\|_\infty$ as we do in the next lemma.

Lemma 4.8. *For all $t > m_1$, the bound*

$$\|\mathbb{E}_{D,A_2} [\tilde{\mathbf{x}}_{t,r}^2]\|_\infty \leq 4 \|\mathbb{E}_D [\mathbf{x}^2]\|_1 + \frac{20}{3} d\epsilon$$

holds with probability 1 if $\epsilon = 1$ and with probability $\geq 1 - \delta$, if $\epsilon \leq 1$.

The proof can be found in Appendix B.12.

In the lasso scenario it is sufficient to use one bound (compare to Lemma 3.7 in the ridge scenario) as we are able to join the two regimes of ϵ by ensuring $\epsilon \leq 1$ (Algorithm 4, line 11). Using this bound, the proof of the theorem is straightforward. First, using Theorem 4.1 on the second phase of the algorithm, we have

$$\mathbb{E}_{D,A_2} [L_{\mathcal{D}}(\bar{\mathbf{w}})] - L_{\mathcal{D}}(\mathbf{w}^*) \leq B \left(\frac{\log 2d}{\eta m_2} + 5\eta G^2 \right). \quad (9)$$

Now we use Lemma 4.8, plug it into Lemma 4.5 and have $G^2 \leq 4B^2 \left(\frac{4}{k} \|\mathbb{E}_D [\mathbf{x}^2]\|_1 + \frac{20}{3k} d\epsilon + 1 \right)$ with probability 1 if $\epsilon = 1$ and with probability $\geq 1 - \delta$, if $\epsilon \leq 1$. We continue by denoting $\widehat{G}^2 = 4B^2 \left(\frac{4}{k} \|2\mathbf{A} + \frac{10}{3}\epsilon\|_1 + \frac{20}{3k} d\epsilon + 1 \right)$ and by using equation (15) we obtain $G^2 \leq \widehat{G}^2$. Plugging $\eta = \sqrt{\frac{\log 2d}{\widehat{G}^2 5m_2}} = \sqrt{\frac{k \log 2d}{20B^2 m_2 (8\|\mathbf{A}\|_1 + 20d\epsilon + k)}}$ into equation (9), we have

$$\begin{aligned} \mathbb{E}_{D,A_2} [L_{\mathcal{D}}(\bar{\mathbf{w}})] - L_{\mathcal{D}}(\mathbf{w}^*) &\leq B \left(\frac{\log 2d}{m_2 \eta} + 5\eta G^2 \right) \\ &\leq B \left(\frac{\log 2d}{m_2 \eta} + 5\eta \widehat{G}^2 \right) \\ &\leq 2B \sqrt{\frac{5\widehat{G}^2 \log 2d}{m_2}} \\ &\leq 4B^2 \sqrt{\frac{5 \left(4\|2\mathbf{A} + \frac{10}{3}\epsilon\|_1 + \frac{20}{3} d\epsilon + k \right) \log 2d}{k m_2}}. \end{aligned}$$

Using

$$\left\| 2\mathbf{A} + \frac{10}{3}\epsilon \right\|_1 \leq \left\| 4\mathbb{E}_D [\mathbf{x}^2] + \frac{14}{6}\epsilon + \frac{10}{3}\epsilon \right\|_1 \leq 4 \|\mathbb{E}_D [\mathbf{x}^2]\|_1 + \frac{17}{3} d\epsilon, \quad (10)$$

we have

$$\begin{aligned}\mathbb{E}_{D, A_2} [L_{\mathcal{D}}(\bar{\mathbf{w}})] - L_{\mathcal{D}}(\mathbf{w}^*) &\leq 4B^2 \sqrt{\frac{5(16 \|\mathbb{E}_D[\mathbf{x}^2]\|_1 + \frac{68}{3}d\epsilon + \frac{20}{3}d\epsilon + k) \log 2d}{km_2}} \\ &\leq 4B^2 \sqrt{\frac{5(16 \|\mathbb{E}_D[\mathbf{x}^2]\|_1 + \frac{88}{3}d\epsilon + k) \log 2d}{km_2}}.\end{aligned}$$

If $\epsilon = 1$, we have

$$4B^2 \sqrt{\frac{5(16 \|\mathbb{E}_D[\mathbf{x}^2]\|_1 + \frac{88}{3}d\epsilon + k) \log 2d}{km_2}} \leq 61B^2 \sqrt{\frac{d \log 2d}{km_2}}$$

with probability 1. Otherwise plugging in $\epsilon = \min\left(\frac{d \log \frac{2d}{\delta}}{(k+1)m_1}, 1\right)$ finishes the proof.

5 Experiments

In this section we describe some experiments designed to test our algorithms and substantiate our analytical claims. We conducted 3 sets of experiments: on an artificial data set that allows us to easily control the properties of the data such as $\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}}$ and $\|\mathbb{E}_D[\mathbf{x}^2]\|_1$ and to show the dependance of the algorithms on them; on a subset of the popular MNIST [9] data set, containing only the "3" and "5" digits, similar to [3, 4]; and on the Coverttype [10] data set. MNIST and Coverttype were designed for a binary classification task, which was addressed by regressing on the -1 and +1 labels.

For the ridge regression scenario, each test consists of 5 algorithms:

1. Our DDAERR algorithm that has prior knowledge of the second moment of the attributes.
2. Our Two-Phased DDAERR algorithm that does not have prior knowledge of the second moments of the attributes, and tries to estimate them.
3. The AERR algorithm that does not require any prior knowledge.
4. Online ridge regression that performs online gradient descent and has access to all the attributes.
5. Offline ridge regression that minimizes the empirical risk, which also has access to all attributes, and moreover, utilized the data better than the online algorithm, as it uses each training example more than once.

For the lasso scenario we used the corresponding algorithms. In all cases our algorithms used the improved inner product estimation as well as the improved data point estimation, as discussed in page 8.

For a fair comparison between the attribute efficient algorithms and the full-information algorithms, we use the X-axis in our figures to represent the number of attributes each algorithm sees, and not the number of examples that is usually used in these kinds of comparisons. The reason for this is that we would like to compare the algorithms by the total budget they use.

To quantify the theoretical improvement of the DDAERR algorithm, we need to compare $\|\mathbb{E}_D [\mathbf{x}^2]\|_{\frac{1}{2}}$ to d , as this is the potential improvement according to our analysis. To avoid scaling issues, we also normalize by $\mathbb{E}_D [\|\mathbf{x}^2\|_1]$, and define our 'Improvement Ratio' by

$$\rho_{\text{ridge}} = \frac{\|\mathbb{E}_D [\mathbf{x}^2]\|_{\frac{1}{2}}}{d\mathbb{E}_D [\|\mathbf{x}^2\|_1]} \quad (11)$$

We prefer this definition upon a simpler definition using the exact bound ratio of the different algorithms because we want to emphasize that this quantity is a property of the data set itself, and is not algorithm nor analysis dependent.

Similarly, for the lasso scenario, we define

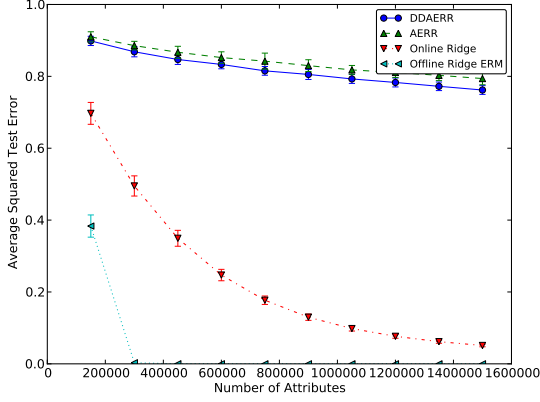
$$\rho_{\text{lasso}} = \frac{\|\mathbb{E}_D [\mathbf{x}^2]\|_1}{d\|\mathbb{E}_D [\mathbf{x}^2]\|_{\infty}} \quad (12)$$

For each data set and algorithm we have used 10-fold cross validation, similar to [3, 4], to optimize the parameters for each phase, and run the learning process 100 times on increasingly long prefixes of the training set to obtain a sense of the variability of the results. We measured the performance of each algorithm by the average loss over the testing set, divided by the loss of the zero predictor, and defined the error bars as one standard deviation. For the two-phased algorithms, we set $m_1 = \frac{m}{10}$, $m_2 = \frac{9m}{10}$, and run the AERR/AELR algorithm during the first phase, using its result as a starting point for the second phase. Unlike the theoretical analysis, we set ϵ to be 0, as the theoretical upper confidence bound is conservative and we found that this improves the empirical results (though increases their variability). We have also split the attribute budget evenly between the data point estimation and the inner product estimation, as it improved the empirical results as well.

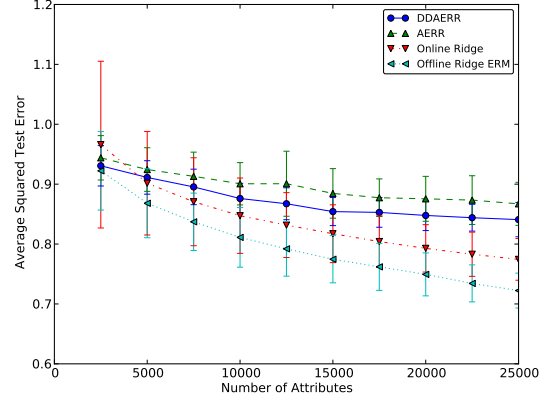
5.1 Simulated Data

We begin by studying a synthetic linear regression data set that easily allows us to control the improvement ratio in both scenarios and to demonstrate the dependence of the algorithms on them. For each experiment, we first defined a vector $\mathbf{u} \in \mathbb{R}^d$ for $d = 500$ by an exponent decaying factor: $u_i = i^\alpha$ for some $\alpha \leq 0$ and then projected the vector on the L_2 ball of radius 1 for the ridge scenario (and on the L_∞ ball of radius 1 for the lasso scenario), to produce the expected values of each attribute, namely the vector $\mathbb{E}[\mathbf{x}]$. To generate one training example, we generated independent binary variables with the corresponding expectations, and joined them into one d -dimensional vector. To generate the entire training set, we repeated the example generation process independently m times, where in each experiment we used a different m to emphasis the interesting regime. For all these experiments, we used $k + 1 = 5$.

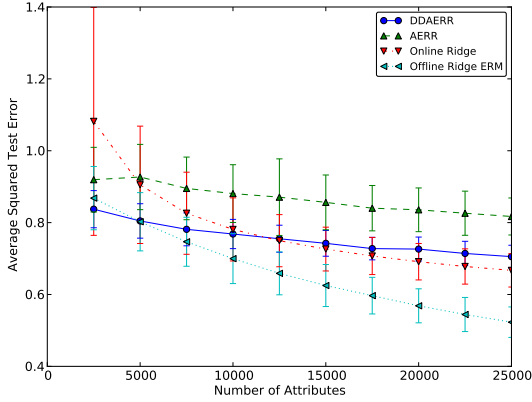
For the ridge scenario, the target values were generated using a scalar product with a random weight vector from $\{-1, 1\}^d$, $\mathbf{w}_{\text{ridge}}^*$, which itself was generated i.i.d. with $P(w_{\text{ridge},i}^* = 1) = P(w_{\text{ridge},i}^* = -1) = 0.5$. For the lasso scenario, the target values were generated using a scalar product with a random sparse weight vector from $\{-1, 0, 1\}^d$, $\mathbf{w}_{\text{lasso}}^*$, which was generated i.i.d. with $P(w_{\text{lasso},i}^* = 1) = P(w_{\text{lasso},i}^* = -1) = 0.15$ and $P(w_{\text{lasso},i}^* = 0) = 0.7$.



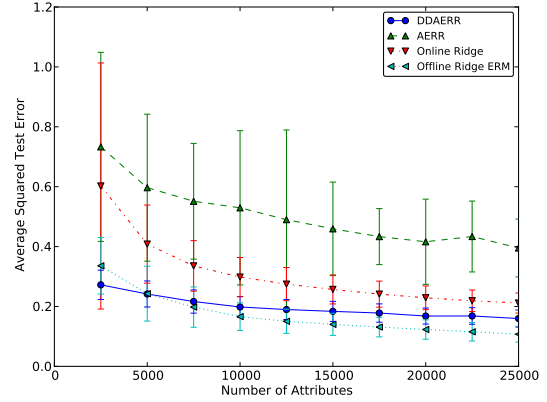
(a) The second moments were chosen to be equal, which results in an improvement factor of $\rho_{\text{ridge}} = 1$.



(b) The second moments were chosen by a power law with an exponent of $\alpha = -0.5$, which results in an improvement factor of $\rho_{\text{ridge}} = 0.91$.



(c) The second moments were chosen by a power law with an exponent of $\alpha = -1$, which results in an improvement factor of $\rho_{\text{ridge}} = 0.55$.



(d) The second moments were chosen by a power law with an exponent of $\alpha = -2$, which results in an improvement factor of $\rho_{\text{ridge}} = 0.05$.

Figure 1: Test error for the algorithms with $k + 1 = 5$ in the ridge scenario over simulated data with $d = 500$.

The results for the ridge scenario appear in figure 1: In the first experiment, all the attributes have the same distribution, therefore we have $\rho_{\text{ridge}} = 1$, and the DDAERR and AERR algorithms are equivalent⁵. As ρ_{ridge} decreases, the algorithms drift apart, and we see a significant improvement in our methods as predicted by our theory.

The results for the lasso scenario that appear figure 2 show the same behaviour, this time with respect to $\|\mathbb{E}_D [\mathbf{x}^2]\|_1$ instead of $\|\mathbb{E}_D [\mathbf{x}^2]\|_{\frac{1}{2}}$.

5.2 MNIST Data Set

In our next set of experiments, we choose to repeat the experiments in [3, 4] and use the popular MNIST data set. Each training example is a labeled 28×28 gray scale image of one hand-written digit. As in the original experiments, we have focused on the classification problem of distinguishing between the "3" digits (which we labeled -1) and the "5" digits (which we labeled +1). As in [4], we have used $k + 1 = 57$ attributes for each training example in the ridge scenario and $k + 1 = 5$ attributes in the lasso scenario. For this data set we have $d = 784$, $\rho_{\text{ridge}} = 0.45$ and $\rho_{\text{lasso}} = 0.2$.

The results for the ridge scenario appear in figure 3: our DDAERR algorithm performs considerably better than the AERR algorithm, for all the training set sizes checked, in correspondence with the theory. Also, the DDAERR algorithm performs similarly to the online ridge algorithm, and even better for a small total number of examined attributes. This suggests that at least for a small number of total attributes, our attribute efficient method is better than the full-information method. The offline ridge algorithm is still the best algorithm, because it can utilize all attributes from each example thus reducing the variance, as well as use each example more than once - privileges the attribute efficient algorithms lack. The Two-Phased DDAERR algorithm performs between the AERR algorithm and the DDAERR algorithm, and converges towards the DDAERR algorithm as the number of observed attributes grows, as expected.

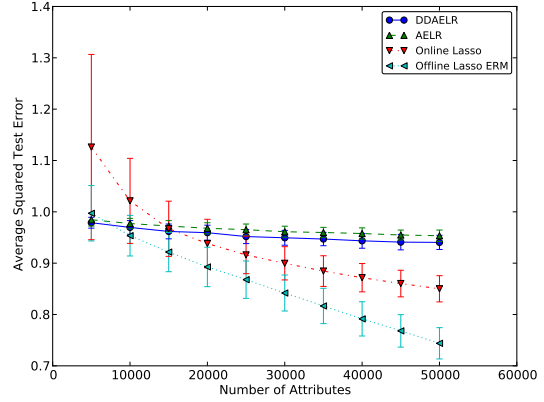
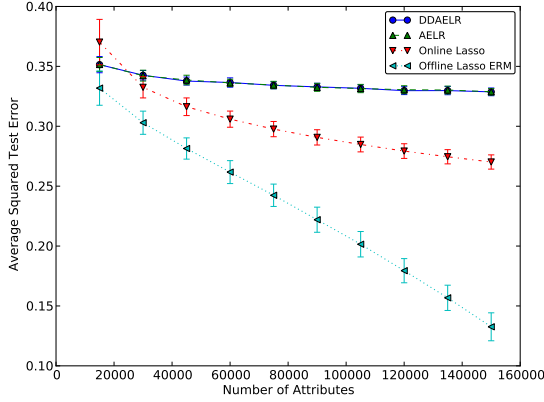
The results for the lasso scenario which appear in figure 4 are similar: The DDAELR algorithm performs considerably better than the AELR algorithm, and comparable with the online lasso algorithm, if not slightly better. It is interesting to note that the variability of the DDAELR algorithm is smaller than the variabilities of the other algorithms. Also, this time it is much clearer that the Two-Phased DDAELR algorithm performs similarly to the AELR algorithm for a small amounts of examined attributes, and converges to DDAELR as the number of examined attributes increases.

5.3 Covertypes Data Set

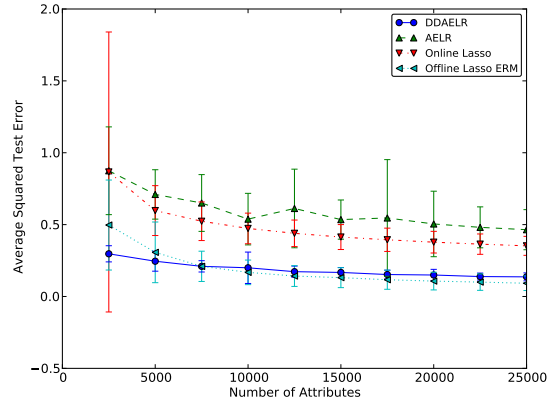
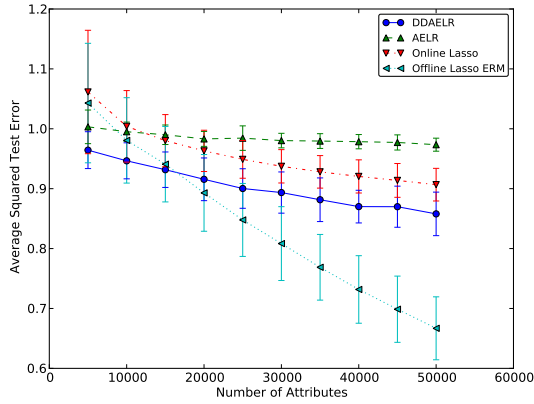
In our last set of experiments we used the Covertypes data set which aims to predict the forest cover type i.e. the dominant species of tree, from cartographic variables. This data set is designed for multi class classification, but we reduce it to a binary classification by choosing one of the tree species and address the problem by regressing on the -1 and +1 labels. For both the ridge and lasso scenarios, we use a budget of $k + 1 = 5$. For this data set we have $d = 54$, $\rho_{\text{ridge}} = 0.49$ and $\rho_{\text{lasso}} = 0.08$.

The results for the ridge scenario appear in figure 5: Again, our DDAERR algorithm performs considerably better than the AERR algorithm. Also, the DDAERR algorithm performs similarly to the online ridge algorithm for a small number of examined attributes. The Two-Phased DDAERR

⁵The small difference between the algorithms is caused by the difference between the methods each algorithm uses when calculating the optimal step size, η .



(a) The second moments were chosen to be equal, (b) The second moments were chosen by a power law with an exponent of $\alpha = -0.5$, which results in an improvement factor of $\rho_{\text{lasso}} = 1$. law with an exponent of $\alpha = -0.5$, which results in an improvement factor of $\rho_{\text{lasso}} = 0.086$.



(c) The second moments were chosen by a power law with an exponent of $\alpha = -1$, which results in an improvement factor of $\rho_{\text{lasso}} = 0.014$. (d) The second moments were chosen by a power law with an exponent of $\alpha = -2$, which results in an improvement factor of $\rho_{\text{lasso}} = 0.0033$.

Figure 2: Test error for the algorithms with $k + 1 = 5$ in the lasso scenario over simulated data with $d = 500$.

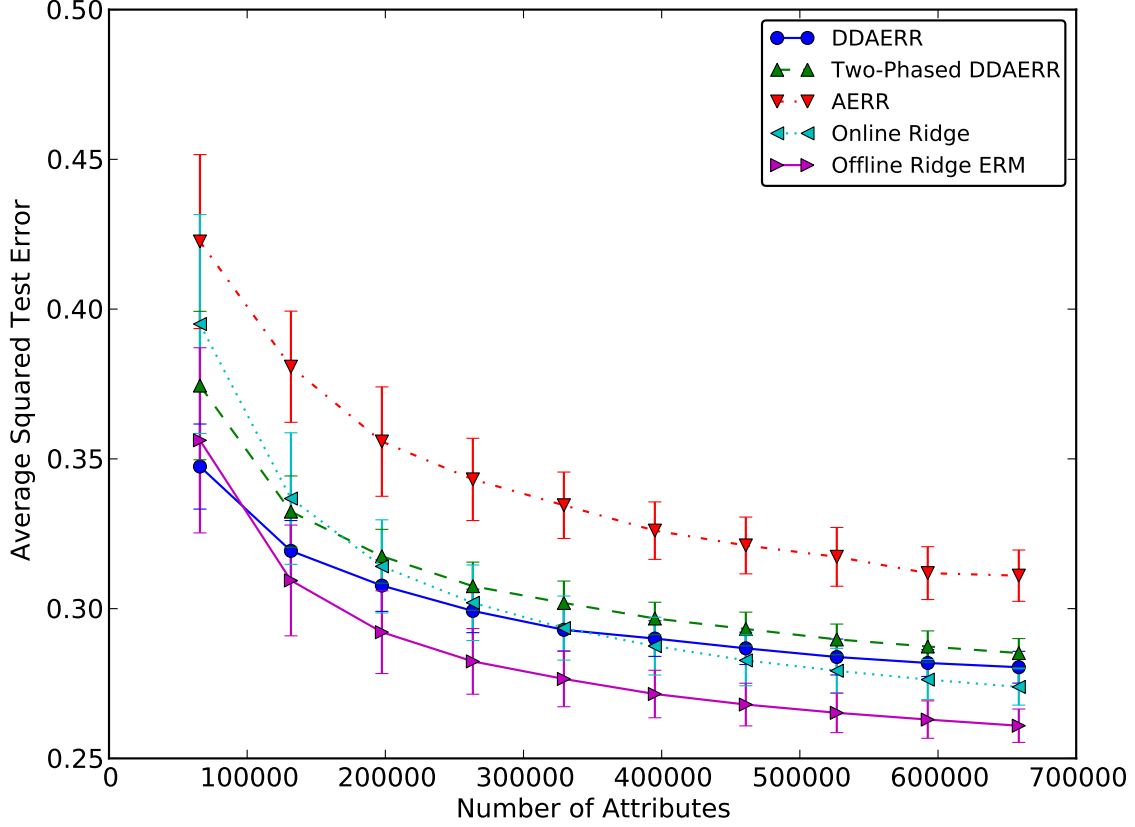


Figure 3: Test error for the algorithms with $k + 1 = 57$ in the ridge scenario over the classification task "3" vs. "5" in the MNIST data set.

algorithm performs between the AERR algorithm and the DDAERR algorithm, and given a larger training set will probably converge towards the DDAERR algorithm as the number of examined attributes grow. This time, however, the full-information ridge algorithms outperform the attribute efficient ones.

The results for the lasso scenario which appear in figure 6 are similar: The DDAELR algorithm performs better than the AELR algorithm. Also, the Two-Phased DDAELR algorithm performs between the AELR and DDAELR algorithms and converges towards the DDAELR algorithm, as the number of attributes grows. For a small number of examined attributes, the DDAELR algorithm performs similarly to the online lasso algorithm, and with a smaller variability, but as the number of examined attributes grow, the algorithms drift apart.

6 Summary and Extensions

In this paper, we studied the attribute efficient local budget setting and developed efficient linear algorithms for the ridge and lasso regression scenarios. Our algorithms utilize the geometry of the

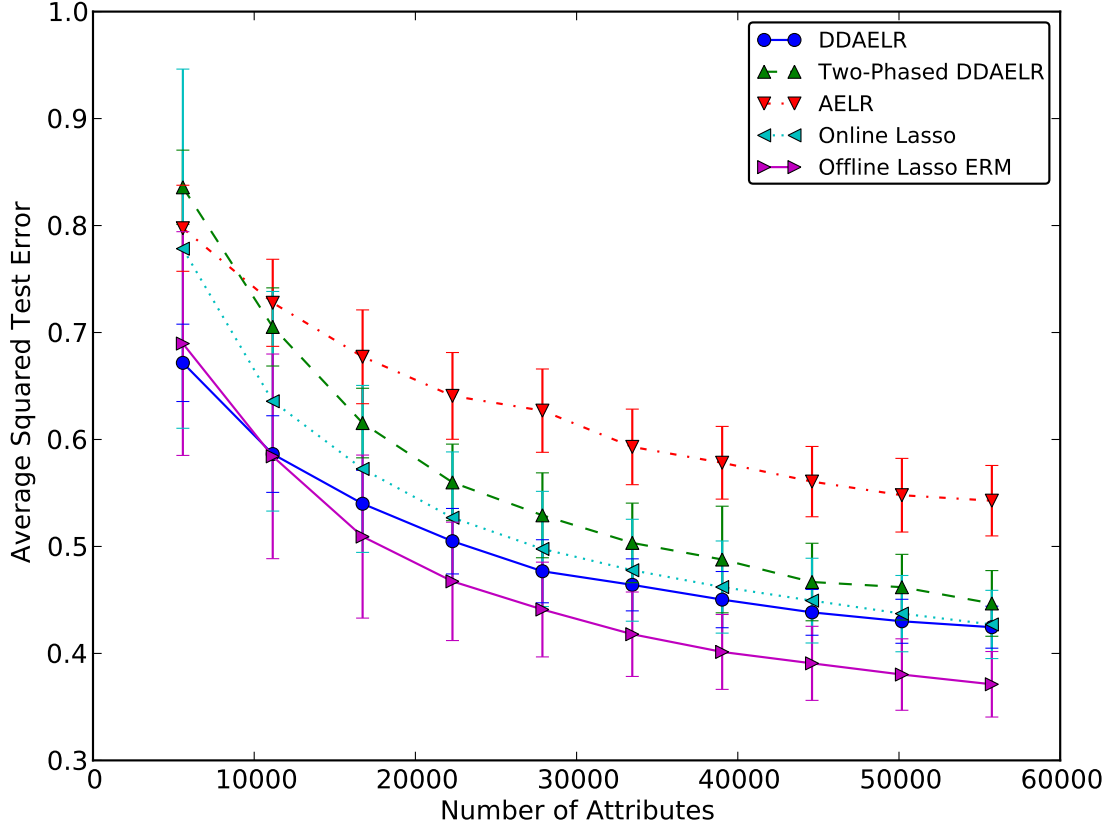


Figure 4: Test error for the algorithms with $k + 1 = 5$ in the lasso scenario over the classification task "3" vs. "5" in the MNIST data set.

data distribution, and are able to achieve data-dependent improvement factors for the excess risk bound over the state of the art, which can be large as $O(\sqrt{d})$. We proved our claims analytically as well as demonstrated them empirically over several data sets.

Our method, even though applied here only for regression scenarios, is quite general, and potentially will be effective in other partial-information learning problems.

There are several possible directions for further research: First, as our algorithm bounds hold only in expectation, the question of how to extend them to hold with high probability, arises. Second, while our work focuses on learning from i.i.d. stochastic data, it is interesting to understand whether analogous results can hold in the online learning scenario [11], where the data is not assumed to be stochastic. In addition, understanding the exact connection between the attribute efficient algorithms and the adaptive methods may lead to an additional improvement in our algorithms. Another direction for future research may be to use the geometry of the optimal linear predictor besides the geometry of the data. For example, if for some i , $\mathbf{w}_{t,i}$ is small, perhaps the learner should sample it less. Finally, proving data-dependent lower bounds may complement our results, or show additional room for improvement.

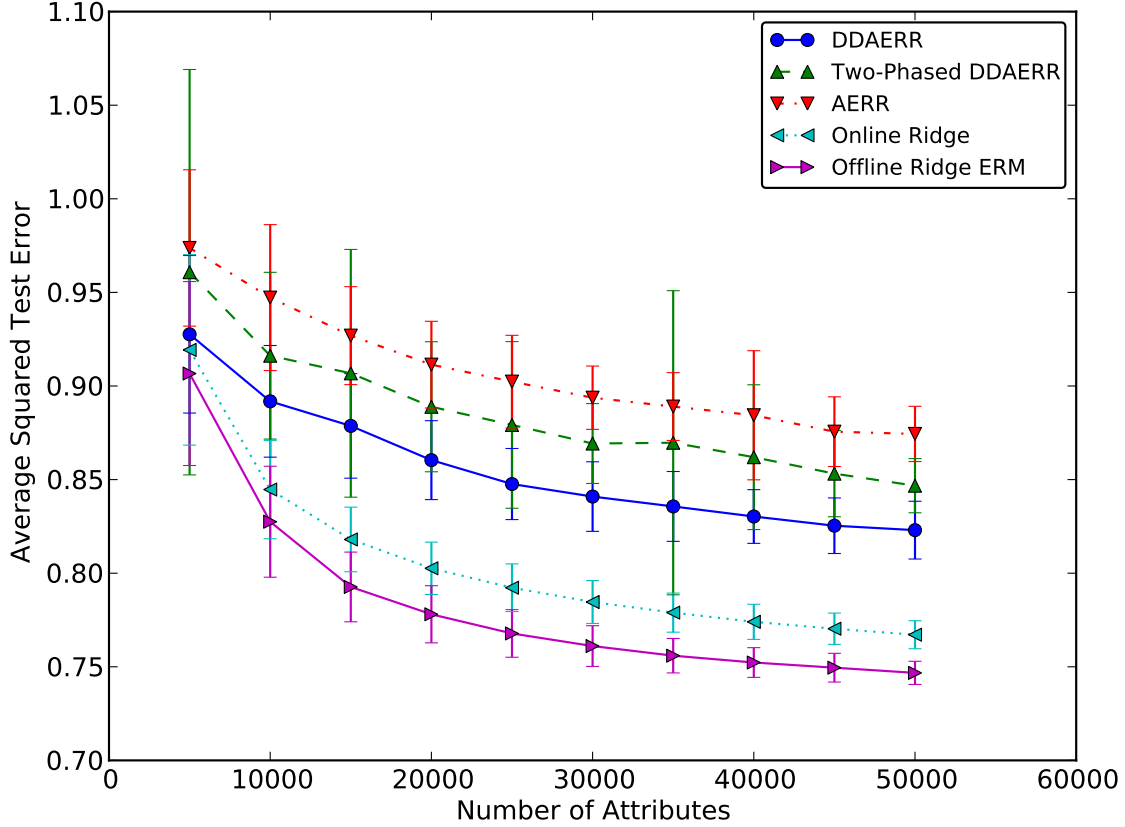


Figure 5: Test error for the algorithms with $k + 1 = 5$ in the ridge scenario over the classification task in the Cover Type data set.

Acknowledgements

This research was partially supported by an Israel Science Foundation Grant (425/13) and an FP7 Marie Curie CIG grant.

References

- [1] Omid Madani, Daniel J Lizotte, and Russell Greiner. Active model selection. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 357–365. AUAI Press, 2004.
- [2] Shai Ben-David and Eli Dichterman. Learning with restricted focus of attention. In *Proceedings of the sixth annual conference on Computational learning theory*, pages 287–296. ACM, 1993.
- [3] Nicolo Cesa-Bianchi, Shai Shalev-Shwartz, and Ohad Shamir. Efficient learning with partially observed attributes. *The Journal of Machine Learning Research*, 12:2857–2878, 2011.

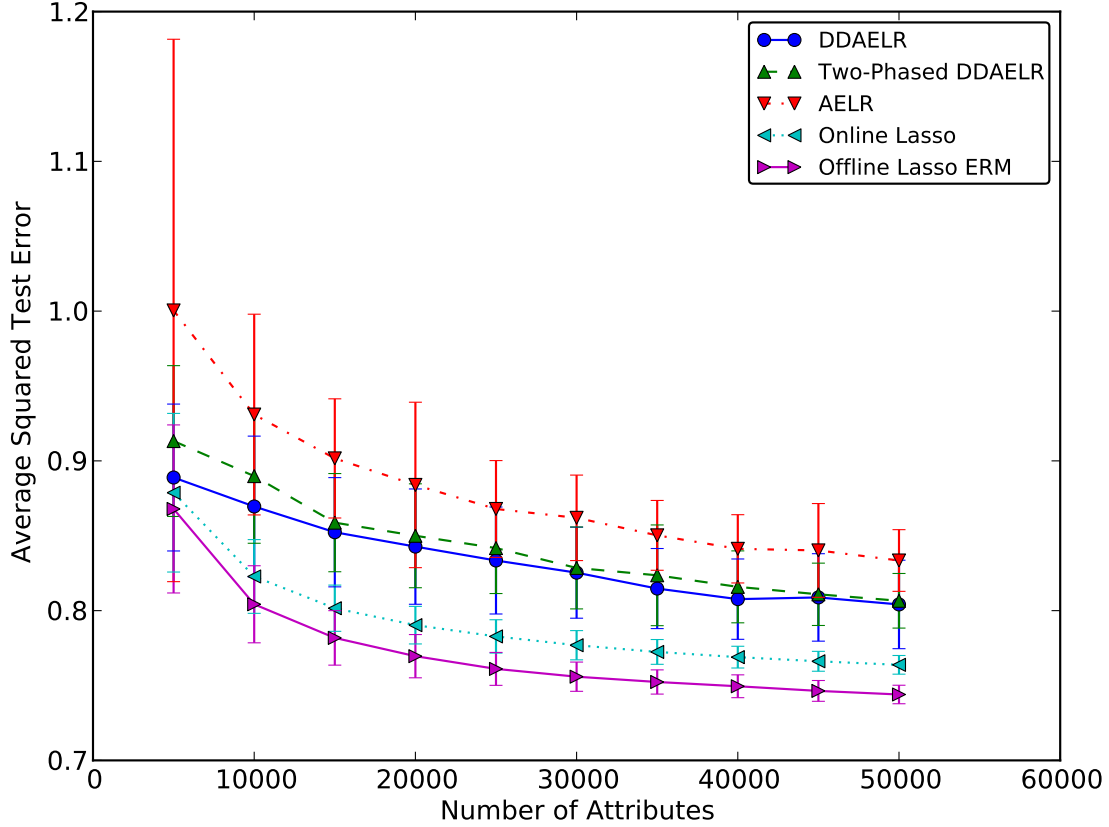


Figure 6: Test error for the algorithms with $k + 1 = 5$ in the lasso scenario over the classification task in the Cover Type data set.

- [4] Elad Hazan and Tomer Koren. Optimal algorithms for ridge and lasso regression with partially observed attributes. *arXiv preprint arXiv:1108.4559*, 2011.
- [5] Sham M Kakade, Karthik Sridharan, and Ambuj Tewari. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. In *Advances in neural information processing systems*, pages 793–800, 2009.
- [6] David Haussler. Decision theoretic generalizations of the pac model for neural net and other learning applications. *Information and computation*, 100(1):78–150, 1992.
- [7] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. 2003.
- [8] Jyrki Kivinen and Manfred K Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1–63, 1997.
- [9] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

- [10] Jock A Blackard and Denis J Dean. Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. *Computers and electronics in agriculture*, 24(3):131–151, 1999.
- [11] Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2011.
- [12] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011.
- [13] Kenneth L Clarkson, Elad Hazan, and David P Woodruff. Sublinear optimization for machine learning. *Journal of the ACM (JACM)*, 59(5):23, 2012.

A AdaGrad

Our data-dependent results rely on the data having different moments along different directions. Such a situation has also been used to improve gradient descent methods in the full information case. The idea is that in such cases, it may be more beneficial to use a different learning rate along each coordinate, rather than a single global rate.

In particular, AdaGrad [12] is a popular approach along these lines. Instead of using a global step size, it uses a different step size for each coordinate, defined as

$$\eta_{t,i} = \text{O} \left(\frac{1}{\sqrt{\sum_{s=1}^t \tilde{g}_{s,i}^2}} \right) \quad (13)$$

for the i -th. The algorithm can be considered as running a separate copy of OGD for each attribute, with a suitable learning rate.

The algorithm is never worse than using a global learning rate, and may be better by a factor dependent on the variability of the magnitude of the gradient estimates across attributes, similarly to our data-dependent algorithms. Therefore, a question arises of what is the connection between our algorithm and AdaGrad, and whether they interfere or support each other.

In this paper we do not theoretically analyze an adaptive gradient version of our algorithms, but provide and discuss simulation results.

We run two simulations. The first, on the artificial data set from section 5.1 with $d = 500$, a decaying exponent of $\alpha = -2$ and $\rho_{\text{ridge}} = 0.05$. As in the original experiment, we have used $k + 1 = 5$. The second experiment, on the subset of the MNIST data set from section 5.2 with $d = 784$ and $\rho_{\text{ridge}} = 0.45$. As in the original experiment, we have used $k + 1 = 57$.

The results for the simulated data set appear in figure 7: In all cases except the offline ERM, the adaptive algorithm performs slightly better than the corresponding non-adaptive algorithm. For the offline ERM algorithm, both perform the same.

The results for the MNIST data set, which appear in figure 8, are different: The adaptive version of the full-information algorithms performs slightly better than the corresponding full-information algorithms. However, the adaptive version of the attribute efficient algorithms performs worse than their non-adaptive version.

The bottom line of these simulations is not conclusive: In some scenarios, the adaptive method improve the results whereas in others, it degrades them. Further research is required in order to

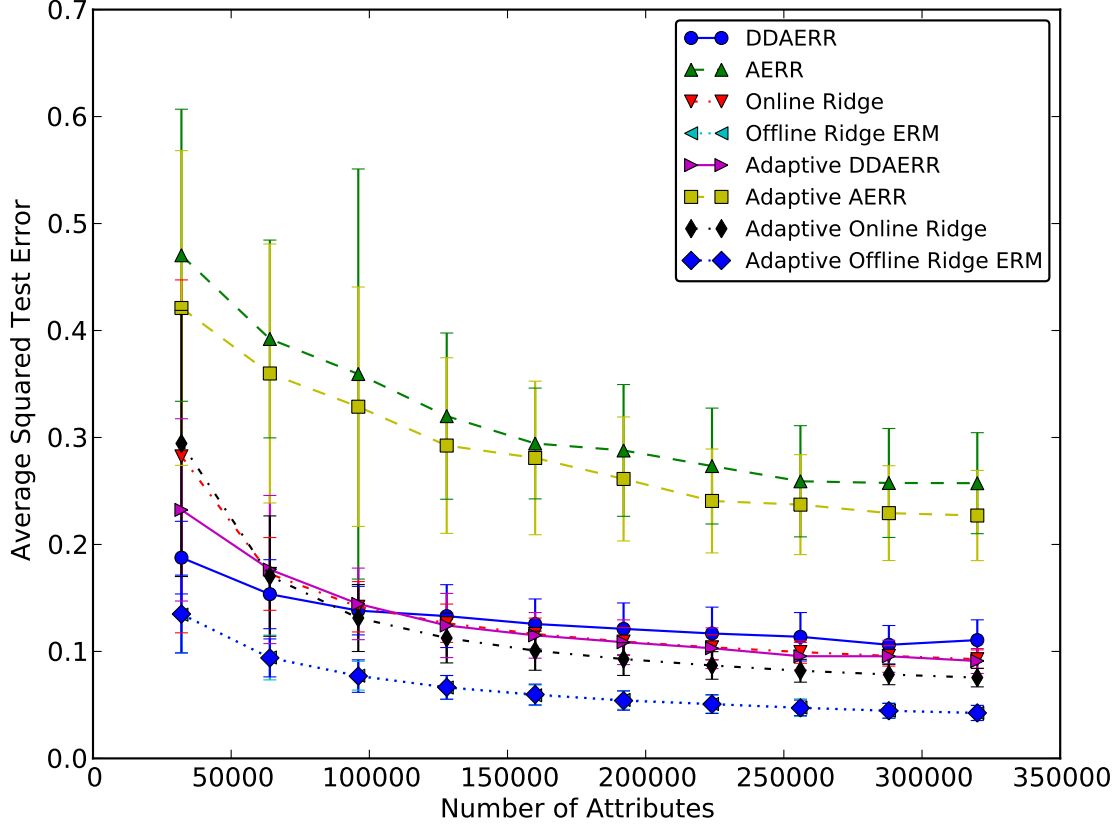


Figure 7: Test error for the algorithms with $k + 1 = 5$ in the ridge scenario over simulated data with $d = 500$, a decaying exponent of $\alpha = -2$ and $\rho_{\text{ridge}} = 0.05$.

understand better the connection between the attribute efficient scenario and the adaptive gradient descent improvements.

B Proofs

B.1 Proof of Theorem 3.1

We use the standard analysis of the OGD algorithm. The expected risk bound of the it is stated in the following lemma.

Lemma B.1 (Zinkevich, 2003). *For any $\|\mathbf{w}^*\| \leq B$, we have*

$$\sum_{t=1}^m \tilde{\mathbf{g}}_t^T (\mathbf{w}_t - \mathbf{w}^*) \leq \frac{2B^2}{\eta} + \frac{\eta}{2} \sum_{t=1}^m \|\tilde{\mathbf{g}}_t\|_2^2. \quad (14)$$

The proof can be found in [7].

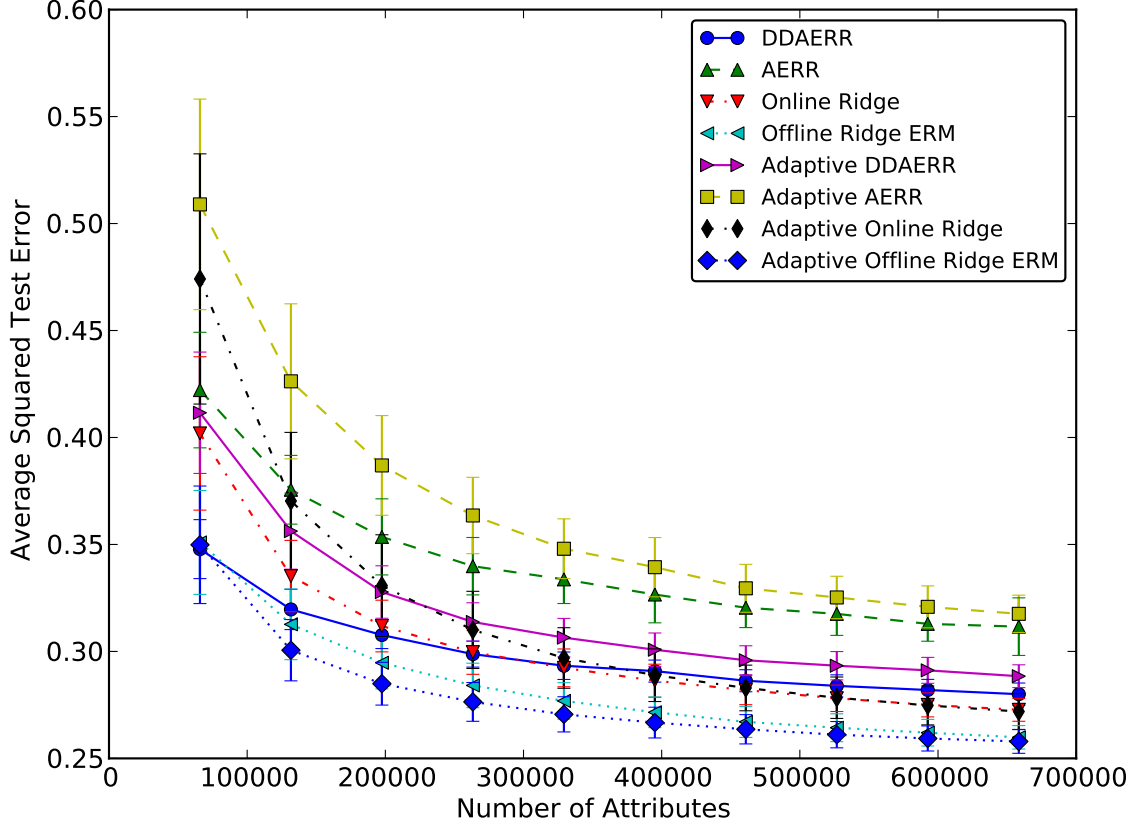


Figure 8: Test error for the algorithms with $k + 1 = 57$ in the ridge scenario over the classification task "3" vs. "5" in the MNIST data set.

To use this lemma, first we need to prove that the GAERR algorithm actually performs stochastic gradient descent. To show this, it is enough to prove that $\tilde{\mathbf{g}}_t$ is an unbiased estimator of the gradient, as stated in the next lemma:

Lemma B.2. *The vector $\tilde{\mathbf{g}}_t$ is an unbiased estimator of the gradient $\mathbf{g}_t = (\mathbf{w}_t^T \mathbf{x}_t - y_t) \mathbf{x}_t$, that is $\mathbb{E}_A [\tilde{\mathbf{g}}_t] = \mathbf{g}_t$.*

Now, we can take the expectation of Lemma B.1 with respect to the randomization of the algorithm and the data distribution, and using Lemma B.2 we have

$$\mathbb{E}_{D,A} \left[\sum_{t=1}^m \mathbf{g}_t^T (\mathbf{w}_t - \mathbf{w}^*) \right] \leq \frac{2B^2}{\eta} + \frac{\eta}{2} G^2 m.$$

On the other hand, the convexity of ℓ gives $\ell_t(\mathbf{w}_t) - \ell_t(\mathbf{w}^*) \leq \mathbf{g}_t^T (\mathbf{w}_t - \mathbf{w}^*)$. Together with the above we have

$$\mathbb{E}_{D,A} \left[\frac{1}{m} \sum_{t=1}^m \ell_t(\mathbf{w}_t) \right] \leq \mathbb{E}_{D,A} \left[\frac{1}{m} \sum_{t=1}^m \ell_t(\mathbf{w}^*) \right] + \frac{2B^2}{\eta m} + \frac{\eta}{2} G^2,$$

or

$$\mathbb{E}_{D,A} \left[\frac{1}{m} \sum_{t=1}^m L_{\mathcal{D}}(\mathbf{w}_t) \right] \leq L_{\mathcal{D}}(\mathbf{w}^*) + \frac{2B^2}{\eta m} + \frac{\eta}{2} G^2,$$

Using the convexity of $L_{\mathcal{D}}$ and Jensen's inequality, the theorem follows.

Proof of Lemma B.2. First, it is straightforward to see that $\mathbb{E}_A[\tilde{\mathbf{x}}_{t,r}] = \mathbf{x}_t$ for all r thus also $\mathbb{E}_A[\tilde{\mathbf{x}}_t] = \mathbf{x}_t$. Also, a simple calculation shows that

$$\mathbb{E}_A[\tilde{\phi}_t] = \sum_{j=1}^d p_j \left(\frac{w_{t,j}}{p_j} \mathbf{x}_t[j] - y_t \right) = \mathbf{w}_t^T \mathbf{x}_t - y_t.$$

Since $\tilde{\mathbf{x}}_t$ and $\tilde{\phi}_t$ are independent given \mathbf{x}_t , we obtain that $\mathbb{E}_A[\tilde{\mathbf{g}}_t] = (\mathbf{w}_t^T \mathbf{x}_t - y_t) \cdot \mathbf{x}_t$, which is the required gradient. \square

B.2 Proof of Lemma 3.2

From the definition of $\tilde{\mathbf{x}}_t$ in equation (1),

$$\begin{aligned} \mathbb{E}_{D,A} [\|\tilde{\mathbf{x}}_t\|_2^2] &= \frac{1}{k^2} \mathbb{E}_{D,A} \left[\left\| \sum_{r=1}^k \tilde{\mathbf{x}}_{t,r} \right\|_2^2 \right] \\ &= \frac{1}{k^2} \sum_{r=1}^k \mathbb{E}_{D,A} [\|\tilde{\mathbf{x}}_{t,r}\|_2^2] + \frac{1}{k^2} \sum_{r=1}^k \sum_{s \neq r}^k \mathbb{E}_{D,A} [\langle \tilde{\mathbf{x}}_{t,r}, \tilde{\mathbf{x}}_{t,s} \rangle]. \end{aligned}$$

Since $\mathbb{E}_{D,A}[\tilde{\mathbf{x}}_{t,r}] = \mathbb{E}_D[\mathbf{x}]$ and $\tilde{\mathbf{x}}_{t,r}$ and $\tilde{\mathbf{x}}_{t,s}$ are independent of each other, we finally have

$$\mathbb{E}_{D,A} [\|\tilde{\mathbf{x}}_t\|_2^2] = \frac{1}{k} \mathbb{E}_{D,A} [\|\tilde{\mathbf{x}}_{t,r}\|_2^2] + \frac{k^2 - k}{k^2} \mathbb{E}_D [\|\mathbf{x}\|_2]^2 = \frac{1}{k} \mathbb{E}_{D,A} [\|\tilde{\mathbf{x}}_{t,r}\|_2^2] + \frac{k-1}{k} \mathbb{E}_D [\|\mathbf{x}\|_2]^2.$$

B.3 Proof of Lemma 3.3

Recalling $|y_t| \leq B$ and using the inequality $(a - b)^2 \leq 2(a^2 + b^2)$, by a straightforward calculation we obtain

$$\begin{aligned} \mathbb{E}_{D,A} [\tilde{\phi}_t^2] &= \mathbb{E}_{D,A} \left[\left(\frac{w_{t,j}}{p_j} \mathbf{x}_t[j] - y_t \right)^2 \right] \\ &\leq 2 \mathbb{E}_{D,A} \left[\left(\frac{w_{t,j}}{p_j} \mathbf{x}_t[j] \right)^2 + y_t^2 \right] \\ &\leq 2 \sum_{j=1}^d \frac{1}{p_j} w_{t,j}^2 \mathbb{E}_D [\mathbf{x}_j^2] + 2B^2 \\ &= 2 \|\mathbf{w}_t\|_2^2 \mathbb{E}_D [\|\mathbf{x}\|_2^2] + 2B^2 \\ &\leq 4B^2. \end{aligned}$$

B.4 Proof of Lemma 3.7

First, we state a simple probabilistic lemma that will be used to bound the our estimates for the second moment of the attributes.

Lemma B.3. *Let Z_1, Z_2, \dots, Z_n be i.i.d random variables. $Z_i \in [0, 1]$. Let $\hat{\mathbb{E}}[Z] = \frac{1}{n} \sum_{i=1}^n Z_i$ be their average. Then, with probability $\geq 1 - \delta$*

$$\hat{\mathbb{E}}[Z] \leq 2\mathbb{E}[Z] + \frac{7 \log \frac{1}{\delta}}{6n}.$$

Also, with probability $\geq 1 - \delta$

$$\hat{\mathbb{E}}[Z] \geq \frac{1}{2}\mathbb{E}[Z] - \frac{5 \log \frac{1}{\delta}}{3n}.$$

We prefer to use this lemma rather than the standard Bernstein inequality because we are interested in a fast convergence rate of $\frac{1}{n}$, and are willing to pay the price of the additional constant factor.

To prove our lemma, we use the definition of $\|\tilde{\mathbf{x}}_{t,r}\|_2^2$,

$$\mathbb{E}_{D,A_2} [\|\tilde{\mathbf{x}}_{t,r}\|_2^2] = \mathbb{E}_{D,A_2} [\tilde{\mathbf{x}}_{t,r} [i_{t,r}]^2] = \sum_{i=1}^d \frac{1}{q_i} \mathbb{E}_D [x_i^2] = \sum_{j=1}^d \sqrt{A[j] + \frac{13}{6}\epsilon} \sum_{i=1}^d \frac{\mathbb{E}_D [x_i^2]}{\sqrt{A[i] + \frac{13}{6}\epsilon}}.$$

For all $i \in [d]$ let T_i be a random variable describing the amount of times the algorithm sampled the i -th attribute in the first phase. For every realization t_i of T_i , since T_i and the samples themselves are independent, we can use Lemma B.3 and by the union bound have that with probability larger than $1 - \delta$, $\mathbf{A}[i] \leq 2\mathbb{E}_D [x_i^2] + \frac{7}{6}\mathbb{E}_{A_1} [\epsilon_i]$, and $\mathbf{A}[i] \geq \frac{1}{2}\mathbb{E}_D [x_i^2] - \frac{5}{3}\mathbb{E}_{A_1} [\epsilon_i]$ where $\epsilon_i = \frac{\log \frac{2d}{\delta}}{t_i}$. Clearly, $\mathbb{E}_{A_1} [T_i] = \frac{(k+1)m_1}{d}$, and using the convexity of $f(x) = \frac{1}{x}$ we have $\mathbb{E}_{A_1} [\epsilon_i] \geq \frac{d \log \frac{2d}{\delta}}{(k+1)m_1} = \epsilon$. Therefore, with probability $\geq 1 - \delta$ over the first phase, we have

$$\begin{cases} \mathbf{A}[i] \leq 2\mathbb{E}_D [x_i^2] + \frac{7}{6}\epsilon \\ \mathbf{A}[i] \geq \frac{1}{2}\mathbb{E}_D [x_i^2] - \frac{5}{3}\epsilon. \end{cases} \quad (15)$$

Note that these equations also hold trivially for any $\epsilon \geq 1$ as with probability 1 we have $x_i^2 \leq 1$ for all $i \in [d]$.

Now we can continue and see,

$$\begin{aligned} \mathbb{E}_{D,A_2} [\|\tilde{\mathbf{x}}_{t,r}\|_2^2] &\leq \sum_{j=1}^d \sqrt{2\mathbb{E}_D [x_j^2] + \frac{7}{6}\epsilon + \frac{13}{6}\epsilon} \sum_{i=1}^d \frac{\mathbb{E}_D [x_i^2]}{\sqrt{\frac{1}{2}\mathbb{E}_D [x_i^2] - \frac{5}{3}\epsilon + \frac{13}{6}\epsilon}} \\ &= \sum_{j=1}^d \sqrt{2\left(\mathbb{E}_D [x_j^2] + \frac{5}{3}\epsilon\right)} \sum_{i=1}^d \frac{\mathbb{E}_D [x_i^2]}{\sqrt{\frac{1}{2}\left(\mathbb{E}_D [x_i^2] + \epsilon\right)}} \\ &= 2 \sum_{j=1}^d \sqrt{\mathbb{E}_D [x_j^2] + \frac{5}{3}\epsilon} \sum_{i=1}^d \frac{\mathbb{E}_D [x_i^2]}{\sqrt{\mathbb{E}_D [x_i^2] + \epsilon}}. \end{aligned}$$

We shall bound this value in two ways. For the first part of the lemma, we have

$$\begin{aligned}
\mathbb{E}_{D,A_2} \left[\|\tilde{\mathbf{x}}_{t,r}\|_2^2 \right] &\leq 2 \sum_{j=1}^d \sqrt{\mathbb{E}_D [x_j^2]} + \frac{5}{3}\epsilon \sum_{i=1}^d \frac{\mathbb{E}_D [x_i^2]}{\sqrt{\mathbb{E}_D [x_i^2]} + \epsilon} \\
&\leq 2 \sum_{j=1}^d \sqrt{\mathbb{E}_D [x_j^2]} \sum_{i=1}^d \frac{\mathbb{E}_D [x_i^2]}{\sqrt{\mathbb{E}_D [x_i^2]} + \epsilon} + 2 \sum_{j=1}^d \sqrt{\frac{5}{3}\epsilon} \sum_{i=1}^d \frac{\mathbb{E}_D [x_i^2]}{\sqrt{\mathbb{E}_D [x_i^2]} + \epsilon} \\
&\leq 2 \sum_{j=1}^d \sqrt{\mathbb{E}_D [x_j^2]} \sum_{i=1}^d \frac{\mathbb{E}_D [x_i^2]}{\sqrt{\mathbb{E}_D [x_i^2]}} + 2 \sum_{j=1}^d \sqrt{\frac{5}{3}\epsilon} \sum_{i=1}^d \frac{\mathbb{E}_D [x_i^2]}{\sqrt{\epsilon}} \\
&\leq 2 \|\mathbb{E}_D [\mathbf{x}^2]\|_{\frac{1}{2}} + 2d\sqrt{\frac{5}{3}} \sum_{i=1}^d \mathbb{E}_D [x_i^2] \\
&\leq 2 \|\mathbb{E}_D [\mathbf{x}^2]\|_{\frac{1}{2}} + 2\sqrt{\frac{5}{3}}d \\
&\leq 5d.
\end{aligned}$$

As this bound is independent of ϵ , it actually holds with probability 1 over the first phase.

For the second part of the lemma, we have

$$\begin{aligned}
\mathbb{E}_{D,A_2} \left[\|\tilde{\mathbf{x}}_{t,r}\|_2^2 \right] &\leq 2 \sum_{j=1}^d \sqrt{\mathbb{E}_D [x_j^2]} + \frac{5}{3}\epsilon \sum_{i=1}^d \frac{\mathbb{E}_D [x_i^2]}{\sqrt{\mathbb{E}_D [x_i^2]} + \epsilon} \\
&\leq 2 \sum_{j=1}^d \sqrt{\mathbb{E}_D [x_j^2]} + \frac{5}{3}\epsilon \sum_{i=1}^d \frac{\mathbb{E}_D [x_i^2]}{\sqrt{\mathbb{E}_D [x_i^2]}} \\
&\leq 2 \sum_{j=1}^d \sqrt{\mathbb{E}_D [x_j^2]} \sum_{i=1}^d \frac{\mathbb{E}_D [x_i^2]}{\sqrt{\mathbb{E}_D [x_i^2]}} + 2 \sum_{j=1}^d \sqrt{\frac{5}{3}\epsilon} \sum_{i=1}^d \frac{\mathbb{E}_D [x_i^2]}{\sqrt{\mathbb{E}_D [x_i^2]}} \\
&\leq 2 \|\mathbb{E}_D [\mathbf{x}^2]\|_{\frac{1}{2}} + 2\sqrt{\frac{5}{3}}d\sqrt{\|\mathbb{E}_D [\mathbf{x}^2]\|_{\frac{1}{2}}}\sqrt{\epsilon}.
\end{aligned}$$

Proof of Lemma B.3. Let us denote the variance of Z by $\sigma^2 = \mathbb{E} [Z^2] - \mathbb{E} [Z]^2$. Using Bernstein's inequality, with probability $\geq 1 - \delta$, we have

$$\hat{\mathbb{E}} [Z] \leq \mathbb{E} [Z] + \sqrt{\frac{2\sigma^2 \log \frac{1}{\delta}}{n}} + \frac{2 \log \frac{1}{\delta}}{3n}.$$

Using $Z_i \in [0, 1]$, we obtain $\sigma^2 = \mathbb{E} [Z^2] - \mathbb{E} [Z]^2 \leq \mathbb{E} [Z^2] \leq \mathbb{E} [Z]$. Plugging back in the expression for $\hat{\mathbb{E}} [Z]$,

$$\hat{\mathbb{E}} [Z] \leq \mathbb{E} [Z] + \sqrt{\frac{2\mathbb{E} [Z] \log \frac{1}{\delta}}{n}} + \frac{2 \log \frac{1}{\delta}}{3n}.$$

Using the fact that the geometric mean is smaller or equal to the arithmetic mean, we have

$$\hat{\mathbb{E}} [Z] \leq \mathbb{E} [Z] + \frac{2\mathbb{E} [Z]}{2} + \frac{\log \frac{1}{\delta}}{2n} + \frac{2 \log \frac{1}{\delta}}{3n}$$

or,

$$\hat{\mathbb{E}}[Z] \leq 2\mathbb{E}[Z] + \frac{7 \log \frac{1}{\delta}}{6n},$$

which concludes the first part of the proof.

Similarly, using Bernstein's inequality again, with probability $\geq 1 - \delta$, we have

$$\hat{\mathbb{E}}[Z] \geq \mathbb{E}[Z] - \sqrt{\frac{2\sigma^2 \log \frac{1}{\delta}}{n}} - \frac{2 \log \frac{1}{\delta}}{3n}.$$

Using $\sigma^2 \leq \mathbb{E}[Z]$, this turns to

$$\hat{\mathbb{E}}[Z] \geq \mathbb{E}[Z] - \sqrt{\frac{2\mathbb{E}[Z] \log \frac{1}{\delta}}{n}} - \frac{2 \log \frac{1}{\delta}}{3n}.$$

Again using the fact that the geometric mean is smaller or equal to the arithmetic mean, we have

$$\hat{\mathbb{E}}[Z] \geq \mathbb{E}[Z] - \frac{\mathbb{E}[Z]}{2} - \frac{2 \log \frac{1}{\delta}}{2n} - \frac{2 \log \frac{1}{\delta}}{3n}$$

or,

$$\hat{\mathbb{E}}[Z] \geq \frac{1}{2}\mathbb{E}[Z] - \frac{5 \log \frac{1}{\delta}}{3n},$$

which concludes the proof. \square

B.5 Proof of Lemma 3.8

First, using Theorem 3.1 on the second phase of the algorithm, we have

$$\mathbb{E}_{D, A_2}[L_{\mathcal{D}}(\bar{\mathbf{w}})] - L_{\mathcal{D}}(\mathbf{w}^*) \leq \frac{2B^2}{\eta m_2} + \frac{\eta}{2}G^2. \quad (16)$$

Now we use the first part of Lemma 3.7, plug it into Lemma 3.4 and obtain that with probability 1, we have $G^2 \leq 4B^2 \left(\frac{5d}{k} + 1\right) \leq 24B^2 \frac{d}{k}$. Plugging $\eta = \sqrt{\frac{k}{6dm_2}}$ into equation (16) finishes the proof.

B.6 Proof of Lemma 3.9

We use second part of Lemma 3.7, plug it into Lemma 3.4 and obtain that with probability $\geq 1 - \delta$, we have $G^2 \leq 4B^2 \left(\frac{2}{k} \|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}} + \frac{2}{k} \sqrt{\frac{5}{3}} d \sqrt{\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}}} \sqrt{\epsilon} + 1\right)$. We denote $\widehat{G^2} = 4B^2 \left(\frac{2}{k} H + \frac{2}{k} \sqrt{\frac{5}{3}} d \sqrt{H} \sqrt{\epsilon} + 1\right)$. Since $H \geq \|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}}$ we have $G^2 \leq \widehat{G^2}$. Plugging $\eta = \frac{2B}{\sqrt{\widehat{G^2} m_2}} = \frac{1}{\sqrt{m_2 \left(\frac{2}{k} H + \frac{2}{k} \sqrt{\frac{5}{3}} d \sqrt{H} \sqrt{\epsilon} + 1\right)}}$ into equation (16), we get

$$\begin{aligned}
\mathbb{E}_{D,A_2} [L_{\mathcal{D}}(\bar{\mathbf{w}})] - L_{\mathcal{D}}(\mathbf{w}^*) &\leq \frac{2B^2}{\eta m_2} + \frac{\eta}{2} G^2 \\
&\leq \frac{2B^2}{\eta m_2} + \frac{\eta}{2} \widehat{G^2} \\
&\leq \frac{2B}{\sqrt{m_2}} \sqrt{\widehat{G^2}} \\
&= \frac{4B^2}{\sqrt{m_2}} \sqrt{\frac{2}{k} H + \frac{2}{k} \sqrt{\frac{5}{3}} d \sqrt{H} \sqrt{\epsilon} + 1}.
\end{aligned}$$

B.7 Proof of Lemma 3.10

First, we state a simple lemma that will allow us to combine two risk bounds, each is achieved by a different value of η .

Lemma B.4. *Let $f(\eta) = \frac{A}{\eta} + \eta B G^2$ for some positive constants A, B, G , where $G \leq \min(G_1, G_2)$. Let $\eta_i = \frac{1}{G_i} \sqrt{\frac{A}{B}}$ for $i = 1, 2$. Then $f(\max(\eta_1, \eta_2)) \leq \min(f(\eta_1), f(\eta_2))$.*

By Lemma 3.8, using $\eta = \sqrt{\frac{k}{12dm_2}}$, we have with probability 1,

$$\mathbb{E}_{D,A_2} [L_{\mathcal{D}}(\bar{\mathbf{w}})] - L_{\mathcal{D}}(\mathbf{w}^*) \leq 4B^2 \sqrt{\frac{6d}{km_2}}.$$

Similarly, by Lemma 3.9, using $\eta = \frac{1}{\sqrt{m_2 \left(\frac{2}{k} H + \frac{2}{k} \sqrt{\frac{5}{3}} d \sqrt{H} \sqrt{\frac{d \log \frac{2d}{\delta}}}{(k+1)m_1} + 1 \right)}}$, we have with probability $\geq 1 - \delta$,

$$\mathbb{E}_{D,A_2} [L_{\mathcal{D}}(\bar{\mathbf{w}})] - L_{\mathcal{D}}(\mathbf{w}^*) \leq \frac{4B^2}{\sqrt{m_2}} \sqrt{\frac{2}{k} H + \frac{2}{k} \sqrt{\frac{5}{3}} d \sqrt{H} \sqrt{\frac{d \log \frac{2d}{\delta}}{(k+1)m_1} + 1}}.$$

Using Theorem 3.1, the expected risk bound has the form of the function in Lemma B.4, and the theorem follows directly.

Proof of Lemma B.4. Assume without loss of generality that $G_1 \geq G_2$, therefore we also have $\eta_2 > \eta_1$. It is enough to prove $f(\eta_2) \leq f(\eta_1)$ which follows directly by simple algebraic manipulations. \square

B.8 Proof of Theorem 4.1

Our analysis is based on the analysis in [4] and brought here for completeness. First, we state the second-order bound for the EG algorithm.

Lemma B.5 (simplified version of Lemma II.3 of [13]). *Let $\eta > 0$, and let $\mathbf{c}_1, \dots, \mathbf{c}_t$ be an arbitrary sequence of vectors in \mathbb{R}^n , with $\mathbf{c}_t[i] \geq -\frac{1}{\eta}$ for all t and all $i \in [n]$. Define a sequence $\mathbf{z}_1, \dots, \mathbf{z}_T$ by letting $\mathbf{z}_1 = \mathbf{1}_n$ and for $t \geq 1$,*

$$\mathbf{z}_{t+1}[i] = \mathbf{z}_t[i] \cdot \exp(-\eta \mathbf{c}_t[i]) \quad i = 1, \dots, n.$$

Then, for the vectors $\mathbf{p}_t = \frac{\mathbf{z}'_t}{\|\mathbf{z}'_t\|_1}$ we have

$$\sum_{t=1}^m \mathbf{p}_t^T \mathbf{c}_t \leq \min_{i \in [n]} \sum_{t=1}^m \mathbf{c}_t[i] + \frac{\log n}{\eta} + \eta \sum_{t=1}^m \mathbf{p}_t^T \mathbf{c}_t^2.$$

Now we examine the vectors $\mathbf{z}' = (\mathbf{z}_t^+, \mathbf{z}_t^-) \in \mathbb{R}^{2d}$ and $\bar{\mathbf{g}}'_t = (\bar{\mathbf{g}}, -\bar{\mathbf{g}}) \in \mathbb{R}^{2d}$, and setting $\mathbf{p}_t = \frac{\mathbf{z}'_t}{\|\mathbf{z}'_t\|_1}$. We have the following lemma:

Lemma B.6 (Lemma 3.5 of [4]).

$$\sum_{t=1}^m \mathbf{p}_t^T \bar{\mathbf{g}}'_t \leq \min_{i \in [2d]} \sum_{t=1}^m \bar{\mathbf{g}}'_t[i] + \frac{\log 2d}{\eta} + \eta \sum_{t=1}^m \mathbf{p}_t^T (\bar{\mathbf{g}}'_t)^2.$$

Using this lemma, we establish an expected risk bound with respect to the clipped linear functions $\bar{\mathbf{g}}_t^T \mathbf{w}$:

Lemma B.7 (Lemma 3.6 of [4]). *Assume that $\|\mathbb{E}_{D,A} [\tilde{\mathbf{g}}_t^2]\|_\infty \leq G^2$ for all t , for some $G \geq 0$. Then, for any $\|\mathbf{w}^*\|_1 \leq B$, we have*

$$\mathbb{E}_{D,A} \left[\sum_{t=1}^m \bar{\mathbf{g}}_t^T \mathbf{w}_t \right] \leq \mathbb{E}_{D,A} \left[\sum_{t=1}^m \bar{\mathbf{g}}_t^T \mathbf{w}^* \right] + B \left(\frac{\log 2d}{\eta} + \eta G^2 m \right).$$

For the proof of Lemma B.9 we will need a simple lemma, that allows us to bound the deviation of the expected value of a clipped random variable from that of the original variable, in terms of its variance.

Lemma B.8. *Let X be a random variable with $|\mathbb{E}[X]| \leq \frac{C}{2}$ for some $C > 0$. Then for the clipped variable $\bar{X} = \text{clip}(X, C) = \max\{\min\{X, C\}, -C\}$ we have*

$$|\mathbb{E}[\bar{X}] - \mathbb{E}[X]| \leq 2 \frac{\text{Var}[X]}{C}.$$

The next step is to relate the risk generated by the linear functions $\tilde{\mathbf{g}}_t^T \mathbf{w}$, to that generated by the clipped functions, $\bar{\mathbf{g}}_t^T \mathbf{w}$.

Lemma B.9 (A correction of Lemma 3.7 of [4]). *Assume that $\|\mathbb{E}[\tilde{\mathbf{g}}_t^2]\|_\infty \leq G^2$ for all t , for some $G \geq 0$. Then, for $0 \leq \eta \leq \frac{1}{2G}$, we have*

$$\mathbb{E}_{D,A} \left[\sum_{t=1}^m \tilde{\mathbf{g}}_t^T (\mathbf{w}_t - \mathbf{w}^*) \right] \leq \mathbb{E}_{D,A} \left[\sum_{t=1}^m \bar{\mathbf{g}}_t^T (\mathbf{w}_t - \mathbf{w}^*) \right] + 4B\eta G^2 m.$$

Using these lemmas, we proceed to the proof of the theorem. First, from Lemma B.2, as the GAERR and GAELR algorithm build the gradient estimator using the same method, we have $\mathbb{E}_A[\tilde{\mathbf{g}}_t] = \mathbf{g}_t$. From this follows that $\mathbb{E}_A[\sum_{t=1}^m \tilde{\mathbf{g}}_t^T (\mathbf{w}_t - \mathbf{w}^*)] = \mathbb{E}_A[\sum_{t=1}^m \mathbf{g}_t^T (\mathbf{w}_t - \mathbf{w}^*)]$. Combining this with Lemmas B.7 and B.9, for $\eta \leq \frac{1}{2G}$, we have

$$\mathbb{E}_{D,A} \left[\sum_{t=1}^m \mathbf{g}_t^T (\mathbf{w}_t - \mathbf{w}^*) \right] \leq \frac{B \log 2d}{\eta} + 5B\eta G^2 m.$$

Proceeding as in the proof of Theorem 3.1 finishes the proof of Theorem 4.1.

Proof of Lemma B.5. Using the fact that $e^x \leq 1 + x + x^2$, for $x \leq 1$, we have

$$\|\mathbf{z}_{t+1}\|_1 = \sum_{i=1}^n \mathbf{z}_t[i] \cdot e^{-\eta \mathbf{c}_t[i]} \leq \sum_{i=1}^n \mathbf{z}_t[i] \cdot \left(1 - \eta \mathbf{c}_t[i] + \eta^2 \mathbf{c}_t[i]^2\right) = \|\mathbf{z}_t\|_1 \cdot \left(1 - \eta \mathbf{p}_t^T \mathbf{c}_t + \eta^2 \mathbf{p}_t^T \mathbf{c}_t^2\right),$$

and since $e^z \geq 1 + z$ for $z \in \mathbb{R}$, this implies by induction that

$$\log \|\mathbf{z}_{T+1}\|_1 = \log n + \sum_{t=1}^T \log \left(1 - \eta \mathbf{p}_t^T \mathbf{c}_t + \eta^2 \mathbf{p}_t^T \mathbf{c}_t^2\right) \leq \log n - \eta \sum_{t=1}^T \mathbf{p}_t^T \mathbf{c}_t + \eta^2 \sum_{t=1}^T \mathbf{p}_t^T \mathbf{c}_t^2.$$

On the other hand, we have

$$\log \|\mathbf{z}_{T+1}\|_1 = \log \sum_{i=1}^n \prod_{t=1}^T e^{\eta \mathbf{c}_t[i]} \geq \log \prod_{t=1}^T e^{\eta \mathbf{c}_t[i^*]} = -\eta \sum_{t=1}^T \mathbf{c}_t[i^*].$$

Combining these two and rearranging, we obtain

$$\sum_{t=1}^m \mathbf{p}_t^T \mathbf{c}_t \leq \sum_{t=1}^m \mathbf{c}_t[i^*] + \frac{\log n}{\eta} + \eta \sum_{t=1}^m \mathbf{p}_t^T \mathbf{c}_t^2$$

for any i^* , which completes the proof. \square

Proof of Lemma B.6. To see how Lemma B.6 follows from Lemma B.5, note that we can write the update rule of the GAELR algorithm in the terms of the augmented vectors, \mathbf{z}_t and $\bar{\mathbf{g}}'_t$ as follows

$$\mathbf{z}_{t+1}[i] = \mathbf{z}_t[i] \cdot \exp(-\eta \bar{\mathbf{g}}'_t[i]) \quad i = 1, \dots, 2d.$$

That is, \mathbf{z}_{t+1} is obtained from \mathbf{z}_t by a multiplicative update based on the vector $\bar{\mathbf{g}}'_t$. Noticing that $\|\bar{\mathbf{g}}'_t\|_\infty = \|\bar{\mathbf{g}}_t\|_\infty \leq \frac{1}{\eta}$, we see from Lemma B.5 that for any i^* ,

$$\sum_{t=1}^m \mathbf{p}_t^T \bar{\mathbf{g}}'_t \leq \sum_{t=1}^m \bar{\mathbf{g}}'_t[i^*] + \frac{\log 2d}{\eta} + \eta \sum_{t=1}^m \mathbf{p}_t^T (\bar{\mathbf{g}}'_t)^2,$$

where $\mathbf{p}_t = \frac{\mathbf{z}_t}{\|\mathbf{z}_t\|_1}$, which gives the lemma. \square

Proof of Lemma B.7. Notice that by our notation,

$$\sum_{t=1}^m \mathbf{p}_t^T \bar{\mathbf{g}}'_t = \sum_{t=1}^m \frac{(\mathbf{z}_t^+, \mathbf{z}_t^-)^T (\bar{\mathbf{g}}_t, -\bar{\mathbf{g}}_t)}{\|\mathbf{z}_t^+\|_1 + \|\mathbf{z}_t^-\|_1} = \frac{1}{B} \sum_{t=1}^m \mathbf{w}_t^T \bar{\mathbf{g}}_t$$

and

$$\min_i \sum_{t=1}^m \bar{\mathbf{g}}'_t[i] = \min_{\|\mathbf{w}\|_1 \leq B} \frac{1}{B} \sum_{t=1}^m \mathbf{w}^T \bar{\mathbf{g}}_t \leq \frac{1}{B} \sum_{t=1}^m \mathbf{w}^{*T} \bar{\mathbf{g}}_t$$

for any \mathbf{w}^* with $\|\mathbf{w}^*\|_1 \leq B$. Plugging into the bound of Lemma B.6, we get

$$\sum_{t=1}^m \bar{\mathbf{g}}_t (\mathbf{w}_t - \mathbf{w}^*) \leq B \left(\frac{\log 2d}{\eta} + \eta \sum_{t=1}^m \mathbf{p}_t^T (\bar{\mathbf{g}}'_t)^2 \right).$$

Finally, taking the expectation with respect to the randomization of the algorithm and the data distribution, and noticing that $\left\| \mathbb{E}_{D,A} [(\bar{\mathbf{g}}'_t)^2] \right\|_\infty \leq \left\| \mathbb{E}_{D,A} [\bar{\mathbf{g}}_t^2] \right\|_\infty \leq G^2$, the proof is complete. \square

Proof of Lemma B.8. As a first step, note that for $x > C$ we have $x - \mathbb{E}[X] \geq C/2$, so that

$$C(x - C) \leq 2(x - \mathbb{E}[X])(x - C) \leq 2(x - \mathbb{E}[X])^2.$$

Hence, denoting by μ the probability measure of X , we obtain

$$\begin{aligned} |\mathbb{E}[\bar{X}] - \mathbb{E}[X]| &\leq \int_{x < -C} (x + C) d\mu + \int_{x > C} (x - C) d\mu \\ &\leq \int_{x > C} (x - C) d\mu \\ &\leq \frac{2}{C} \int_{x > C} (x - \mathbb{E}[X])^2 d\mu \\ &\leq 2 \frac{\text{Var}[X]}{C}. \end{aligned}$$

Similarly one can prove that $\mathbb{E}[\bar{X}] - \mathbb{E}[X] \geq -2\text{Var}[X]/C$, and the result follows. \square

Proof of Lemma B.9. Notice that $\|\mathbb{E}_{D,A}[\tilde{\mathbf{g}}_t^2]\|_\infty \leq G^2$ implies $\|\mathbb{E}_{D,A}[\tilde{\mathbf{g}}_t]\|_\infty \leq G$ as

$$\|\mathbb{E}_{D,A}[\tilde{\mathbf{g}}_t]\|_\infty^2 = \left\| \mathbb{E}_{D,A}[\tilde{\mathbf{g}}_t]^2 \right\|_\infty \leq \|\mathbb{E}_{D,A}[\tilde{\mathbf{g}}_t^2]\|_\infty.$$

Since $\bar{\mathbf{g}}[i] = \text{clip}(\tilde{\mathbf{g}}[i], 1/\eta)$ and $|\mathbb{E}_{D,A}[\tilde{\mathbf{g}}_t[i]]| \leq G \leq 1/2\eta$ the above lemma implies that

$$|\mathbb{E}_{D,A}[\bar{\mathbf{g}}_t[i]] - \mathbb{E}_{D,A}[\tilde{\mathbf{g}}_t[i]]| \leq 2\eta \mathbb{E}_{D,A}[\tilde{\mathbf{g}}_t[i]^2] \leq 2\eta G^2$$

for all i , which means $\|\mathbb{E}_{D,A}[\tilde{\mathbf{g}}_t - \bar{\mathbf{g}}_t]\|_\infty \leq 2\eta G^2$. Together with $\|\mathbf{w}_t - \mathbf{w}^*\|_1 \leq 2B$, this implies,

$$\mathbb{E}_{D,A}[(\bar{\mathbf{g}}_t - \tilde{\mathbf{g}}_t)^T (\mathbf{w}_t - \mathbf{w}^*)] \leq 4\eta G^2.$$

Summing over $t = 1, \dots, m$, and taking the expectations, we obtain the lemma. \square

B.9 Proof of Lemma 4.2

From the definition of $\tilde{\mathbf{x}}_t$ in equation (1),

$$\begin{aligned} \|\mathbb{E}_{D,A}[\tilde{\mathbf{x}}_t^2]\|_\infty &= \left\| \mathbb{E}_{D,A}[\tilde{\mathbf{x}}_t[i]^2] \right\|_\infty \\ &= \left\| \mathbb{E}_{D,A} \left[\left(\frac{1}{k} \sum_{r=1}^k \tilde{\mathbf{x}}_{t,r}[i] \right)^2 \right] \right\|_\infty \\ &= \left\| \frac{1}{k^2} \sum_{r=1}^k \mathbb{E}_{D,A}[\tilde{\mathbf{x}}_{t,r}^2[i]] + \frac{1}{k^2} \sum_{r \neq s}^k \mathbb{E}_{D,A}[\tilde{\mathbf{x}}_{t,r}[i]^2] \right\|_\infty. \end{aligned}$$

Since $\mathbb{E}_{D,A}[\tilde{\mathbf{x}}_{t,r}[i]] = \mathbb{E}_D[\mathbf{x}[i]]$, $\tilde{\mathbf{x}}_{t,r}[i]$ and $\tilde{\mathbf{x}}_{t,s}[i]$ are independent of each other, and using the triangle inequality, we finally have

$$\|\mathbb{E}_{D,A}[\tilde{\mathbf{x}}_t^2]\|_\infty \leq \max_i \frac{1}{k} \mathbb{E}_{D,A}[\tilde{\mathbf{x}}_{t,r}^2[i]] + \frac{k-1}{k} \mathbb{E}_D[\|\mathbf{x}\|_\infty]^2.$$

B.10 Proof of Lemma 4.3

Let $C_i = \frac{\mathbb{E}_D[x_i^2]}{q_i}$. Note that $q_i = \frac{\mathbb{E}_D[x_i^2]}{\sum_{j=1}^d \mathbb{E}_D[x_j^2]}$ if, and only if, all C_i are equal. Assume by contradiction that all C_i are not equal, yet they still yield the minimal value for $\max_i \frac{1}{q_i} \mathbb{E}_D[x_i^2]$. Let $I = \{i | C_i = \max_j C_j\}$, and i_0 be an index for which $C_{i_0} < \max_j C_j$, which exists, by our assumption. For $\Delta > 0$, consider a new set of q'_i -s, such that $q'_{i_0} = q_{i_0} - \Delta$, and $q'_i = q_i + \frac{\Delta}{|I|}$ for $i \in I$. For a small enough Δ , still $C'_{i_0} < \max_j C'_j$. Note that this is still a valid assignment of probabilities because $\sum_{i=1}^d q'_i = 1$ and all $q'_i > 0$ for a small enough Δ . However, $\max_j C'_j$ is smaller than $\max_j C_j$, in contradiction to the assumption. Therefore, all C_i are equal and the minimal value is attained when $q_i = \frac{\mathbb{E}_D[x_i^2]}{\sum_{j=1}^d \mathbb{E}_D[x_j^2]}$.

B.11 Proof of Lemma 4.4

Recalling $|y_t| \leq B$ and using the inequality $(a - b)^2 \leq 2(a^2 + b^2)$, by a straightforward calculation we obtain:

$$\begin{aligned} \mathbb{E}_{D,A} [\tilde{\phi}_t^2] &= \mathbb{E}_{D,A} \left[\left(\frac{w_{t,j}}{p_j} \mathbf{x}_t[j_t] - y_t \right)^2 \right] \\ &\leq 2 \mathbb{E}_{D,A} \left[\left(\frac{w_{t,j}}{p_j} \mathbf{x}_t[j_t] \right)^2 + y_t^2 \right] \\ &\leq 2 \sum_{j=1}^d \frac{1}{p_j} w_{t,j}^2 \mathbb{E}_D[\mathbf{x}_j^2] + 2B^2 \\ &\leq 2 \sum_{j=1}^d \frac{\|\mathbf{w}_t\|_1}{|w_{t,j}|} w_{t,j}^2 + 2B^2 \\ &\leq 2 \|\mathbf{w}_t\|_1 \sum_{j=1}^d |w_{t,j}| + 2B^2 \\ &\leq 4B^2. \end{aligned}$$

B.12 Proof of Lemma 4.8

Using the definition of $\|\tilde{\mathbf{x}}_{t,r}\|_2^2$,

$$\|\mathbb{E}_{D,A_2} [\tilde{\mathbf{x}}_{t,r}^2]\|_\infty = \max_i \mathbb{E}_{D,A_2} [\tilde{\mathbf{x}}_{t,r}^2[i]] = \max_i \frac{1}{q_i} \mathbb{E}_D[x_i^2] = \sum_{j=1}^d \left(A[j] + \frac{13}{6}\epsilon \right) \max_i \frac{\mathbb{E}_D[x_i^2]}{A[i] + \frac{13}{6}\epsilon}.$$

Using equations (15), we have

$$\begin{aligned}
\|\mathbb{E}_{D,A_2} [\tilde{\mathbf{x}}_{t,r}^2]\|_\infty &\leq \sum_{j=1}^d \left(2\mathbb{E}_D [x_j^2] + \frac{7}{6}\epsilon + \frac{13}{6}\epsilon \right) \max_i \frac{\mathbb{E}_D [x_i^2]}{\frac{1}{2}\mathbb{E}_D [x_i^2] - \frac{5}{3}\epsilon + \frac{13}{6}\epsilon} \\
&\leq 4 \sum_{j=1}^d \left(\mathbb{E}_D [x_j^2] + \frac{5}{3}\epsilon \right) \max_i \frac{\mathbb{E}_D [x_i^2]}{\mathbb{E}_D [x_i^2] + \epsilon} \\
&\leq 4 \sum_{j=1}^d \left(\mathbb{E}_D [x_j^2] + \frac{5}{3}\epsilon \right) \max_i \frac{\mathbb{E}_D [x_i^2]}{\mathbb{E}_D [x_i^2]} \\
&\leq 4 \|\mathbb{E}_D [\mathbf{x}^2]\|_1 + \frac{20}{3}d\epsilon.
\end{aligned}$$

If $\epsilon = 1$, as equations (15) hold with probability 1, this bound also holds with probability 1. If $\epsilon \leq 1$, this bound holds with probability $\geq 1 - \delta$.